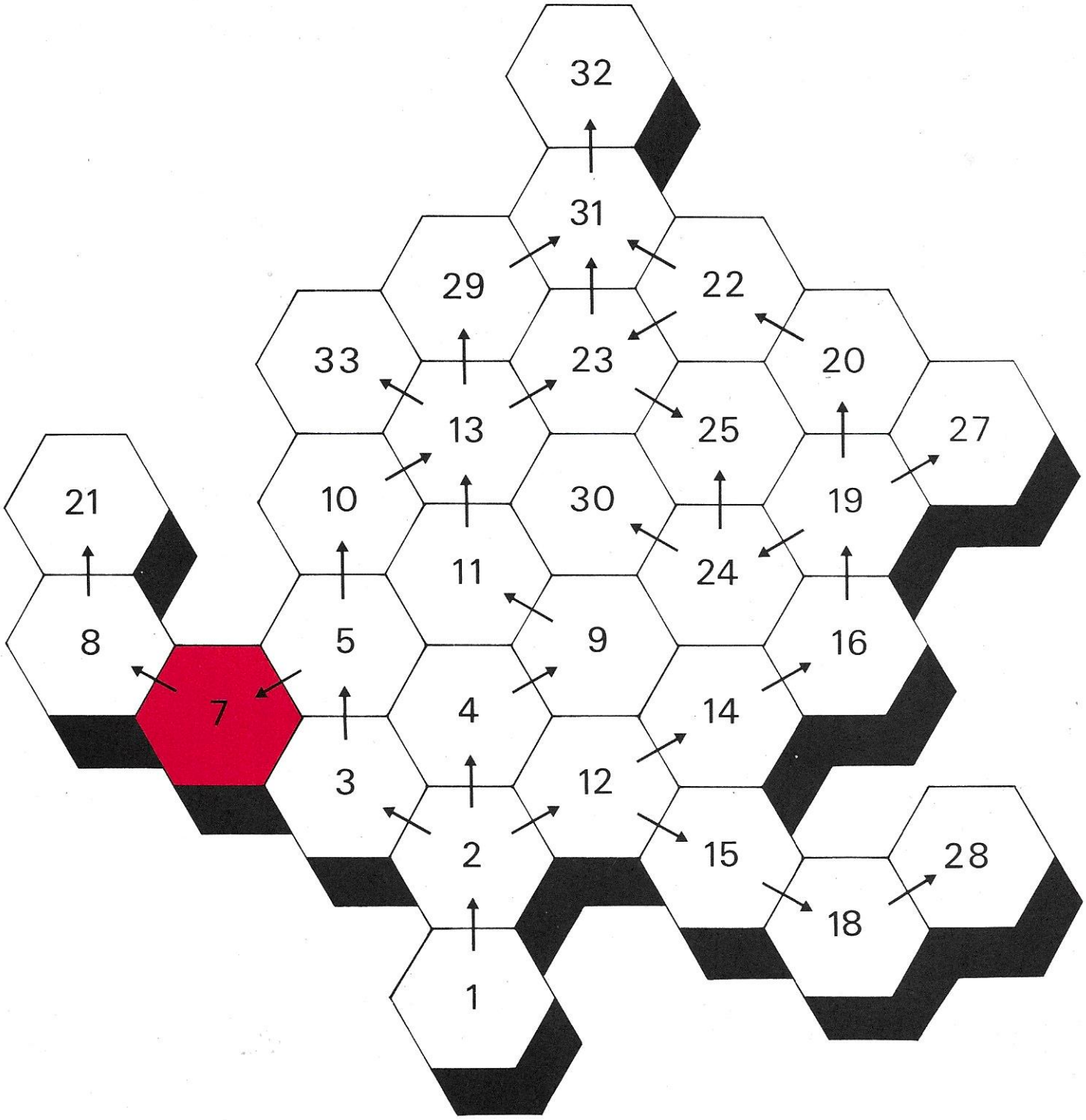
M201 7

THE OPEN UNIVERSITY

Mathematics: A Second Level Course

Linear Mathematics Unit 7

Introduction to Numerical Mathematics: Recurrence Relations



The Open University

Mathematics: A Second Level Course

Linear Mathematics Unit 7

INTRODUCTION TO NUMERICAL MATHEMATICS: RECURRENCE RELATIONS

Prepared by the Linear Mathematics Course Team

The Open University Press

The Open University Press Walton Hall Milton Keynes MK7 6AA

First published 1972. Reprinted 1976

Designed by the Media Development Group of the Open University.

Printed in Great Britain by
Martin Cadbury

SBN 335 01096 2

This text is one in a series of units that make up the correspondence element of an Open University Second Level Course. The complete list of units in the course is given at the end of this text.

For general availability of supporting material referred to in this text, please write to the Director of Marketing, The Open University, P.O. Box 81, Walton Hall, Milton Keynes, MK7 6AT.

Further information on Open University courses may be obtained from the Admissions Office, The Open University, P.O. Box 48, Walton Hall, Milton Keynes, MK7 6AB.

1.2

Contents

Set Books

D. L. Kreider, R. G. Kuller, D. R. Ostberg and F. W. Perkins, *An Introduction to Linear Analysis* (Addison-Wesley, 1966).

E. D. Nering, *Linear Algebra and Matrix Theory* (John Wiley, 1970).

It is essential to have these books; the course is based on them and will not make sense without them.

Bibliography

L. Fox and D. F. Mayers, *Computing Methods for Scientists and Engineers* (Clarendon Press, Oxford 1968). This book will be found to be very useful in connection with the numerical analysis units in this course.

Conventions

Before working through this correspondence text make sure you have read *A Guide to the Linear Mathematics Course*. Of the typographical conventions given in the Guide the following are the most important.

The set books are referred to as:

K for *An Introduction to Linear Analysis*
N for *Linear Algebra and Matrix Theory*

All starred items in the summaries are examinable.

References to the Open University Mathematics Foundation Course Units (The Open University Press, 1971) take the form *Unit M100 3, Operations and Morphisms.*

Note

Please note that this text is not based on the set books for the course.

7.0 INTRODUCTION

The study of mathematics is more than an interesting intellectual exercise. It also has many applications in physical science, engineering and technology, social science, medicine and other fields. Such applications frequently lead to mathematical problems whose solutions are not as easy to calculate as the specially constructed ones you have met so far in this course. For example, it is easy enough to find the inverse of a 3×3 matrix using Hermite normal form—but how would you find the inverse of a 30×30 matrix? The answer "use a computer" doesn't fully deal with the question. What do you tell the computer to do, in order to be sure (a) that the answer the computer gives you is accurate enough and (b) that you are not wasting valuable computer time by using an unnecessarily slow method? The search for answers to such questions has led to the development of a branch of mathematics called *numerical analysis*. In the Foundation Course (M100), we discussed some of the elementary ideas of numerical analysis. In particular, *Unit M100 2, Errors and Accuracy* and *Unit M100 28, Linear Algebra IV* were almost entirely devoted to it. There were also parts of the other units, like *Unit M100 4, Finite Differences*, *Unit M100 9, Integration I*, *Unit M100 14, Sequences and Limits II* and *Unit M100 24, Differential Equations I*, which included a considerable amount of numerical work. Explicit reference will be made to the Foundation Course units where necessary.

In the numerical analysis units of this course we will discuss accurate and economical methods for the calculations in linear mathematics, such as the calculation of matrix inverses, the solution of systems of linear equations, the calculation of eigenvalues and eigenvectors, and the solution of differential equations.

One feature that all these methods have in common is that they depend on arithmetical processes which are repeated many times during the calculation, and which usually introduce small round-off errors at each step. There is a tendency for such errors to accumulate, and sometimes even to be magnified as the calculation proceeds, and in consequence it is quite possible for a badly designed calculation to give wildly wrong answers. For example, you will see later in this unit how a plausible method for the apparently simple problem of evaluating the integral $\int_0^1 x^8 e^{x-1}\,dx$ gives the answer -0.7280 whereas the correct answer is about $+0.1$; the calculation gives an answer with the wrong sign, and about 7 times too big. This disastrous result is caused by the magnification of a small round-off error, less than 0.00005 in magnitude, in the initial data used to start the calculation.

The main purpose of this unit is to show you how such unpleasant possibilities come about and how they can be controlled and avoided. We shall discuss them in the context of the simplest calculation involving a repetitive process—the evaluation of a sequence of numbers from a recurrence formula (or recurrence relation as it is often called), such as the formula $F_n = F_{n-1} + F_{n-2}$ involved in the Fibonacci sequence, which we mentioned in *Unit 5, Determinants and Eigenvalues.*

In sub-section 7.2.3 we explain why the problem of solving a recurrence relation is a linear problem, by considering the vector space whose elements are all sequences of real numbers.

7.1 INSTABILITIES

7.1.0 Introduction

By the term *instability* we refer to any situation where a small error in a number at one stage of the calculation is magnified to give a large error in the answer. There are two main types of instability; one arising from the nature of the problem for which we are trying to compute a solution, the other from the method we use to solve the problem. We consider them in turn.

7.1.1 Ill-conditioned Problems: Inherent Instability

So far in this course we have taken it for granted that, whatever calculation we are doing, we start with well-defined data and that we obtain a well-defined correct result, even though this may be difficult to calculate exactly. Not all calculations, however, are of this type. For example, in many practical problems some of the data have been obtained by the use of measuring apparatus, such as rulers or voltmeters. We saw in *Unit M100 2, Errors and Accuracy* how to represent this situation mathematically: if the result of some measurement is a number x and the accuracy of that measurement is described by the absolute error bound ε, then all we can say about the true value of the measured quantity is that it lies in the interval $[x - \varepsilon, x + \varepsilon]$. Now, the quantity we want to calculate depends on the true value of x, which we may call X; but since we do not know X exactly we cannot find the exact answer. Suppose, for example, that the object of our calculation is to evaluate the image of X under some function f. Then the best we can do is to calculate the image of the interval $[x - \varepsilon, x + \varepsilon]$, and assert that $f(X)$ lies somewhere in this image set. Depending on the nature of the function f and the value of ε, this assertion may or may not provide accurate information about the value of $f(X)$.

Example 1

Suppose a measurement of a physical quantity X gives the value $x = 2.0$ with an accuracy of 2 significant figures. What can be said about the value of X^{10}?

Since we know only that $X \in [1.95, 2.05]$, all we can deduce is that

$$X^{10} \in [(1.95)^{10}, (2.05)^{10}] \simeq [794, 1311]$$

Thus, although we can compute x^{10} as $2^{10} = 1024$, we cannot even be sure that the first digit of this answer is correct.

Example 2

Suppose our measurement is as before, but this time we are interested in $X^{1/10}$. This time we deduce $X^{1/10} \in [(1.95)^{1/10}, (2.05)^{1/10}] \simeq [1.069, 1.075]$, and so the value of $x^{1/10}$, which is 1.072 to 3 decimal places, is in error by at most 0.3%, and its first 3 digits are certainly correct. Thus we should be justified in quoting the answer as $X^{1/10} = 1.07$; but to quote any more figures would be unnecessary and possibly misleading.

Example 3

In the quadratic equation

$$t^2 - 2.029t + 1.028 = 0,$$

suppose that the number 2.029 is known exactly, but that all we know about the number 1.028 is that it is correctly rounded, i.e. that it lies in the interval [1.0275, 1.0285]. Then all we can say about the larger of the

two roots is that it lies between the larger roots of $t^2 - 2.029t + 1.0275 = 0$ and $t^2 - 2.029t + 1.0285 = 0$, i.e. between 1.0412 and 1.0559. Uncertainties in the fifth significant figure of the data produce uncertainties in the third significant figure of the answer.

Three important points emerge from these examples.

(i) Any "rule of thumb" that the accuracy in the answer is roughly the same as that of the data, is quite unfounded.

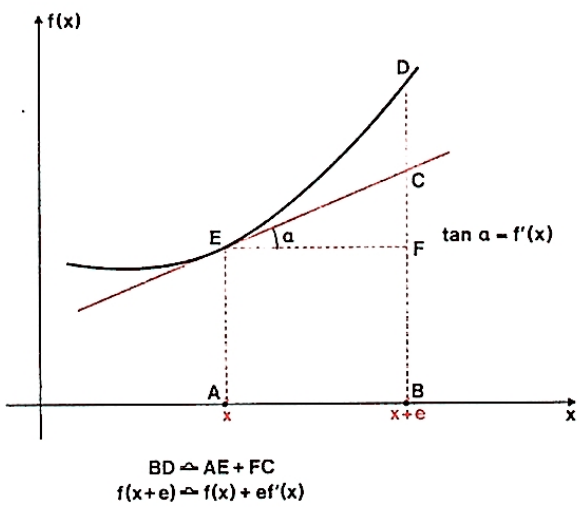
(ii) In Examples 1 and 3 the errors in the answers were much larger than those in the data, whereas in Example 2 we had the opposite effect. We say that problems like Examples 1 and 3, in which any errors in the data are magnified, are *ill-conditioned*, whereas problems in which the errors are diminished are *well-conditioned.*

(iii) This effect depends only on the problem we are attempting and not the method used to calculate its solution. For this reason, ill-conditioning is sometimes referred to as *inherent stability*, to distinguish it from instabilities that are not inherent in the problem itself.

Sometimes we can give a numerical measure of the degree of conditioning. For the problem of computing images under a real function, a natural choice is the *magnification factor* by which an error e in the domain must be multiplied to give the corresponding error in the codomain. If f is differentiable and e is sufficiently small, then

$$f(x + e) \simeq f(x) + ef'(x)$$

(*see Unit M100 14, Sequences and Limits II*).



Thus, the error in the codomain is very close to $ef'(x)$, that is, the magnification factor is very close to the derivative $f'(x)$; but, since we are not interested in the sign of the error, we shall take the magnitude $|f'(x)|$ as our measure of conditioning.

Since it relates absolute rather than relative errors*, we may call $|f'(x)|$ the *absolute condition number* for the problem of calculating $f(X)$. If the absolute condition number is much larger than 1, the problem is said to be (absolutely) ill-conditioned; if much smaller, then it is (absolutely) well-conditioned.

* For absolute and relative errors see *Unit M100 2, Errors and Accuracy.*

An alternative measure of conditioning is the *relative condition number*, defined by using the ratio of the relative rather than the absolute errors in codomain and domain. The relative error in the domain is defined as e/x where e is the error in x, and that in the codomain is approximately $ef'(x)/f(x)$ (for small e); so the relative condition number is

$$\left| \frac{ef'(x)/f(x)}{e/x} \right| = \left| \frac{xf'(x)}{f(x)} \right|.$$

If this number is larger than 1 the problem is said to be relatively ill-conditioned; if smaller, it is relatively well-conditioned.

Although we have introduced the idea of ill-conditioning here in the context of a *physical problem* whose initial data come from an (inaccurate) physical measurement, it is also useful in purely *mathematical* problems, where the problem uses only mathematical data, such as the values of numbers like π or e or $\sqrt{2}$, or even $\frac{1}{3}$. If we look up such numbers in a book of tables, or rely on standard computer subroutines to calculate them, the values we obtain will contain small rounding errors which may be magnified to unacceptable proportions if the problem is ill-conditioned. In this mathematical case, however, we are better off than in the physical problem, because there is no physical limitation on the accuracy of the data—by taking sufficient trouble we can make these mathematical data as accurate as we like. Thus in the case of a physical problem, ill-conditioning imposes absolute restrictions on the accuracy of our result, but in a mathematical problem it is merely a warning that special care will be needed to get high accuracy.

Example 4

Suppose that we wish to compute $y = e^{10}$ for $e = 2.71828\ldots$, and that we wish to have the first three figures correct in the computed result. Taking successive correctly rounded approximations to e, we obtain the rounded values

$$\begin{aligned}
(3)^{10} &= 0.5905 \times 10^5 \\
(2.7)^{10} &= 0.2059 \times 10^5 \\
(2.72)^{10} &= 0.2217 \times 10^5 \\
(2.718)^{10} &= 0.2200 \times 10^5 \\
(2.7183)^{10} &= 0.2203 \times 10^5
\end{aligned}$$

and this last answer is correct to three significant figures. The problem is ill-conditioned, since we must start with at least 5 correct figures in the "data" to be sure of three correct figures in the answer; but because it's a mathematical rather than a physical problem, we are able to get the accuracy we require by increasing the accuracy of the "data" beyond the 3 figures we might have expected before starting the calculation.

Exercises

1. Calculate the absolute and relative condition numbers for Examples 1 and 2. Check that they correctly predict ill-conditioning where it occurs.
2. If y is defined, for any given x (less than $(1.0145)^2$), as the larger solution of the quadratic equation

   $$y^2 - 2.029y + x = 0,$$

   show that the absolute condition number for calculating y from a given value of x is $(2y - 2.029)^{-1}$. Hence show that when $y = 1.028$ the problem is ill-conditioned.
3. Is the mathematical problem of computing $e^{1/10}$, with $e = 2.71828\ldots$ given, ill- or well-conditioned?

Solutions

1. In Example 1 the function is $x \longmapsto x^{10}$ and the value of x is 2. The absolute condition number is $10 \times 2^9 = 5120$ and the relative condition number is $2 \times \dfrac{10 \times 2^9}{2^{10}} = 10$. By either criterion the problem is ill-conditioned.

   In Example 2 the function is $x \longmapsto x^{1/10}$ and the value of x is again 2. The absolute condition number is

$$\frac{1}{10} \times 2^{(1/10)-1} = \frac{2^{1/10}}{20} \simeq \frac{1.072}{20} \simeq 0.05,$$

   and the relative condition number is $\frac{1}{10}$. By either criterion the problem is well-conditioned.

2. The quickest way of getting $f'(x)$, where $y = f(x)$, is as follows. The quadratic can be written

$$[f(x)]^2 - 2.029f(x) + x = 0, \; x < (1.0145)^2$$

   Differentiating, we get

$$2f(x)f'(x) - 2.029f'(x) + 1 = 0$$

   and rearranging, this gives

$$f'(x) = \frac{1}{2.029 - 2f(x)}$$

   Since the solutions are $1.0145 \pm \sqrt{\text{something}}$, the larger solution is greater than 1.0145; so $2f(x)$ exceeds 2.029 and so the absolute condition number is

$$|f'(x)| = \frac{1}{2f(x) - 2.029} = \frac{1}{2y - 2.029}$$

   When $y = 1.028$ the condition number is

$$\frac{1}{2.056 - 2.029} = \frac{1}{0.027} \simeq 37,$$

   so that the problem is ill-conditioned.

3. The absolute condition number is $f'(e)$, where $f : x \longmapsto x^{1/10}$; that is

$$f'(e) = \frac{1}{10} e^{-9/10} \simeq 0.04$$

   and so the problem is well-conditioned.

7.1.2 Induced Instability: Good and Bad Numerical Methods

We now turn to the second main source of instability. We have already mentioned, and we shall demonstrate more clearly in the next section, that computer arithmetic is not exact. Almost every arithmetical operation produces an error, and the combined effect of these errors needs careful analysis. In some methods the build-up of arithmetic errors is so great that the computed result may be a very poor approximation to the true solution even in a well-conditioned mathematical problem, and in a physical problem it may lead to results well outside the true solution interval. This effect is called *induced instability*; do not confuse it with ill-conditioning: ill-conditioning is a feature of the problem, whereas induced instability is a feature of the method used to solve the problem.

It is perhaps a surprizing fact that many of the most obvious numerical methods can exhibit induced instability. We shall see many examples as the course proceeds.

Example 1

We wish to compute, to four significant figures*, both roots of the quadratic equation

$$x^2 - 2x + 0.009 = 0$$

in which all the coefficients are exact. How do we do this using a computer which stores only four decimal digits?

The usual formula for the roots of a quadratic equation gives

$$\begin{aligned} x_1 &= 1 + \sqrt{0.991} \\ &= 1 + 0.9955 \text{ (in our machine)} \\ &= 1.996, \text{ correct to four significant figures.} \\ x_2 &= 1 - \sqrt{0.991} \\ &= 1 - 0.9955 \\ &= 0.0045, \text{ correct to only two significant figures.} \end{aligned}$$

Thus the method has lost two significant figures; it exhibits mild induced instability. How can we get the required 4-figure accuracy in x_2? The reason for the loss of accuracy in the above method is that x_2 was obtained by subtracting two nearly equal quantities. To get 4-figure accuracy we use a different method. A good method is to use the formula for the product of the solutions of a quadratic equation. The product of the solutions of $x^2 + bx + c = 0$ is c, and so, knowing x_1, we have

$$x_2 = 0.009/x_1 = 0.009/1.996 = 0.004\,509.$$

This is almost correct to 4 significant figures (i.e. the absolute error bound of the 4-figure result given is not much greater than 1 in the last decimal place).

Exercises

1. Justify the statement made above, that the result $x_2 = 0.004\,509$ is almost correct to 4 significant figures, by estimating a relative error bound (i.e. a bound on the relative error in x_2).

   (Remember from *Unit M100 2, Errors and Accuracy*, that a relative error bound for a quotient is the sum of the relative error bounds for the numerator and denominator.)

* If necessary, see definition of significant figures in *Unit M100 2, Errors and Accuracy*.

2. Given that $\sqrt{40} = 6.32$ and $\sqrt{41} = 6.40$, correct to 3 significant figures in each case, calculate $\sqrt{41} - \sqrt{40}$

   (a) directly
   (b) by using the fact that

   $(\sqrt{41} - \sqrt{40})(\sqrt{41} + \sqrt{40}) = 1$

   (Use a table of reciprocals.)

   Estimate the relative error in both answers.

   Which method is preferable?

Solutions

1. In the quotient 0.009/1.996, the numerator is exact and the denominator 1.996 has absolute error bound 0.0005 and relative error bound 0.0005/1.996 = 0.000 25 approximately. The relative error bound for the quotient 0.004 509 is therefore 0.000 25 too, and so the absolute error bound is 0.004 509 × 0.000 25. This is less than 0.12×10^{-5}, i.e. not much more than 1 unit in the last significant figure of the computed x_2.

2. (a) $\sqrt{41} - \sqrt{40} = 6.40 - 6.32 = 0.08$

   (b) $$\sqrt{41} - \sqrt{40} = \frac{1}{\sqrt{41} + \sqrt{40}} = \frac{1}{6.40 + 6.32}$$

   $$= \frac{1}{12.72} = 0.078\ 62$$

   In (a) the absolute error bounds for 6.40 and 6.32 are 0.005, so that an absolute error bound for 0.08 is 0.01, corresponding to a relative error bound of $\frac{1}{8}$. In (b) the absolute error bound for 12.72 is 0.01 for the same reason as in (a), but this corresponds to a relative error bound of 0.01/12.72 which is less than 10^{-3}, and so the relative error bound for 0.078 62 is also less than 10^{-3}. Thus method (b) is more than 100 times as accurate as method (a). Once again, subtracting nearly equal numbers leads to a poor result.

7.1.3 Floating-point Number Storage

In order to devise methods of computation that will avoid induced instability and, more generally, to discover how accurate the results of our computation are, we want to be able to analyse the propagation of errors in a computation. These errors arise from the fact that the computer cannot in general carry out arithmetical operations exactly, and this in turn arises from the fact that the computer cannot store numbers exactly. To see how this limitation gives rise to inexact arithmetical operations, let us consider a (hypothetical) four-figure decimal machine working in *floating-point arithmetic*. This means that numbers are stored in the form $10^b \times a$, where b is a positive or negative integer, and $|a|$ is a four-figure number in the interval $[0.1, 1) = \{|a| : 0.1 \leqslant |a| < 1\}$*. For example, 5764 is stored as $10^4 \times 0.5764$, 0.005 764 as $10^{-2} \times 0.5764$, 5.764 as $10^1 \times 0.5764$, and so on. We shall use the symbol $\mathrm{fl}(x)$ to mean our floating-point form of the number x; so that, for example, $\mathrm{fl}(5764) = 10^4 \times 0.5764$.

The first step, of course, is to get the data of the problem into our machine. When any number is fed in it is automatically rounded to four significant digits and then expressed in floating-point form. For example, the number $\frac{1}{3}$ is stored as $10^0 \times 0.3333$, the number $\frac{1}{11} = 0.090\,909\ldots$ as $10^{-1} \times 0.9091$, the number $\pi = 3.141\,59\ldots$ as $10^1 \times 0.3142$, and so on.

Notice that a whole range of numbers will have the same stored value. For example, every number x in the interval [0.576 55, 0.576 65) is stored as $\mathrm{fl}(x) = 10^0 \times 0.5766$†. The maximum error in the decimal part a is $0.00005 = 5 \times 10^{-5}$, so that the maximum absolute error in the stored number $10^b \times a$ is $5 \times 10^{b-5}$. More commonly, we investigate the maximum relative error, and since the stored number is at least 0.1×10^b, it follows that the *relative* error is at most $(5 \times 10^{b-5})/(0.1 \times 10^b)$, that is 5×10^{-4}. Thus, the introduction of a storage error in an item of data x can be described by the formula

$$\mathrm{fl}(x) = x(1 + r_x).$$

The quantity r_x defined by this formula is called the relative error. It depends on x (and may be zero) and has upper bound given by

$$|r_x| \leqslant 5 \times 10^{-4}.$$

Exercises

1. Evaluate r_x in the formula $\mathrm{fl}(x) = x(1 + r_x)$, with (a) $x = \frac{1}{3}$, (b) $x = \frac{1}{11}$, (c) $x = \pi$, for our fictitious four-figure decimal machine, and check that each $|r_x|$ is less than the upper bound given above.
2. In a *binary* machine, the number x is stored in the floating-point form

   $$\mathrm{fl}(x) = 2^b \times a, \quad \text{where} \quad \tfrac{1}{2} \leqslant |a| < 1,$$

   and $|a|$ is stored as a binary number; for example, 0.110 010 1 ... 01. If the machine can store t binary digits for $|a|$, show that

   $$\mathrm{fl}(x) = x(1 + r_x), \quad \text{where} \quad |r_x| \leqslant 2^{-t}.$$

Solutions

1. (a) $10^0 \times 0.3333 = \frac{1}{3}(1 + r_x)$ gives

   $$r = 0.9999 - 1 = -0.0001$$

   (b) $10^{-1} \times 0.9091 = \frac{1}{11}(1 + r_x)$ gives $r_x = 0.000\,01$

* In *Unit M100 8, Computing I* we used a slightly different convention for floating point representation; there $|a|$ was a six-figure number in the interval [1, 10].
† See *Unit M100 2, Errors and Accuracy*, pp. 7–8.

(c) $10^1 \times 0.3142 = \pi(1 + r_x)$ gives

$$r_x \simeq \frac{3.142}{3.141\,59} - 1 = \frac{0.000\,41}{3.141\,59} \simeq 0.000\,13$$

In each case $|r_x| < 5 \times 10^{-4}$.

2. The absolute error bound for $|a|$ is $\frac{1}{2} \times 2^{-t}$ or 2^{-t-1}. The absolute error bound for $2^b \times a$ is therefore 2^{b-t-1}. The relative error bound is

$$\frac{2^{b-t-1}}{2^b \times a} = \frac{2^{-t-1}}{a} \leqslant 2^{-t}$$

since $a \geqslant 2^{-1}$. Thus we have shown that

$|r_x| \leqslant 2^{-t}$.

7.1.4 Floating-point Arithmetic

The next point to examine is how the machine performs arithmetic on its numbers, already stored (in our machine) in four-figure floating-point form. The details vary in different machines, but here we describe the most common methods.

(a) *Addition and Subtraction*

If $x_1 = 10^{b_1} \times a_1$, $x_2 = 10^{b_2} \times a_2$, with $b_1 \geqslant b_2$, then the machine first evaluates

$$10^{b_1}(a_1 \pm 10^{b_2 - b_1} \times a_2)$$

to eight digits and rounds it to four significant digits (in our machine), and then adjusts the exponent (if necessary) to produce the result in standard floating-point form. For example,

(i) $\text{fl}\{10^2 \times 0.5765 + 10^0 \times 0.2946\} = \text{fl}\{10^2(0.5765 + 0.002\,946)\}$
$= \text{fl}\{10^2(0.579\,446)\}$
$= 10^2 \times 0.5794$

(ii) $\text{fl}\{10^2 \times 0.5765 + 10^2 \times 0.4826\} = \text{fl}\{10^2(0.5765 + 0.4826)\}$
$= \text{fl}\{10^2(1.0591)\}$
$= 10^3 \times 0.1059$

(iii) $\text{fl}\{10^2 \times 0.1024 - 10^1 \times 0.9048\} = \text{fl}\{10^2(0.1024 - 0.090\,48)\}$
$= \text{fl}\{10^2(0.011\,92)\}$
$= 10^1 \times 0.1192$

(b) *Multiplication*

With x_1 and x_2 in floating-point form as above, the machine forms $10^{b_1 + b_2}(a_1 \times a_2)$, then rounds the number in brackets to four significant digits (in our machine), and finally adjusts the exponent (if necessary) to give the result in standard floating-point form. For example,

(i) $\text{fl}\{(10^2 \times 0.5765) \times (10^0 \times 0.5765)\}$
$= \text{fl}\{10^2 \times 0.332\,352\,25\}$
$= 10^2 \times 0.3324$

(ii) $\text{fl}\{(10^2 \times 0.5765) \times (10^{-1} \times 0.1111)\}$
$= \text{fl}\{10^1(0.064\,049\,15)\}$
$= 10^0 \times 0.6405$

(c) *Division*

The quotient x_1/x_2 is calculated as follows. If $|a_1| < |a_2|$, the machine calculates a_1/a_2 to eight-figure accuracy, and since the first digit cannot be zero the floating-point result is

$$10^{b_1 - b_2} \times \text{the rounded value of} \frac{a_1}{a_2}.$$

If $|a_1| > |a_2|$, we divide $\frac{1}{10} a_1$ by a_2, with suitable adjustment of the exponent b_1, and the floating-point result again follows easily. For example,

(i) $$\begin{aligned} &\text{fl}\{(10^3 \times 0.5765) \div (10^2 \times 0.6294)\} \\ &\quad = \text{fl}\{10^1 \times 0.915\,951\,70\} \\ &\quad = 10^1 \times 0.9160 \end{aligned}$$

(ii) $$\begin{aligned} &\text{fl}\{(10^3 \times 0.5765) \div (10^{-2} \times 0.4968)\} \\ &\quad = \text{fl}\{(10^4 \times 0.057\,65) \div (10^{-2} \times 0.4968)\} \\ &\quad = \text{fl}\{(10^6 \times 0.116\,042\,67)\} \\ &\quad = 10^6 \times 0.1160 \end{aligned}$$

As in the case of the floating-point storage of data, so here we can obtain an upper bound for the relative error in an arithmetic operation.

Defining this relative error r by

$$\text{fl}(x_1 \circ x_2) = (x_1 \circ x_2)(1 + r),$$

where $\circ$ is $+$, $-$, $\times$ or $\div$, r can be shown, once again, to have the upper bound 5×10^{-4}.

Computers use binary rather than decimal arithmetic. Numbers are stored in the form $2^b \times a$, where $|a|$ is a binary number in the interval $[\frac{1}{2}, 1)$. Our basic error formulas then become $\text{fl}(x) = x(1 + r_x)$, $|r_x| \leqslant 2^{-t}$ (see Exercise 2 of sub-section 7.1.3.), $\text{fl}(x_1 \circ x_2) = (x_1 \circ x_2)(1 + r)$, $|r| \leqslant 2^{-t}$, where t, the so-called "word-length" of the machine, is the number of binary places reserved for the storage of the fractional part of the number.

Various results of floating-point arithmetic conflict with our normal expectations and mathematical ideas. For example, if x_1 and x_2 are positive with $x_1 > x_2$, we expect that $x_1 - x_2$ is also positive but smaller than x_1. In floating-point arithmetic this may not be true. For example, if we use our hypothetical decimal machine to compute $x_1 - x_2$, with $x_1 = 14.195\,01$, $x_2 = 0.004\,97$, the machine first stores

$$\text{fl}(x_2) = 10^{-2} \times 0.4970, \quad \text{fl}(x_1) = 10^2 \times 0.1420,$$

and then performs the floating point subtraction to obtain

$$\begin{aligned} &\text{fl}\{(10^2 \times 0.1420) - (10^{-2} \times 0.4970)\} \\ &\quad = \text{fl}\{10^2(0.1420 - 0.000\,049\,70)\} \\ &\quad = \text{fl}\{10^2(0.141\,950\,30)\} \\ &\quad = 10^2 \times 0.1420 = 14.20, \end{aligned}$$

which is larger than the original x_1!

Moreover, with exact arithmetic there are useful properties like associativity, which ensures that the order of some numerical operations is immaterial:

$$a + (b + c) = (a + b) + c.$$
$$a(bc) = (ab)c.$$

With a succession of floating-point operations, we find that different orderings produce (in general) different results. Although this is not a serious blemish on computer arithmetic, we may, in extreme cases, wish to use the optimum ordering.

Exercise

Evaluate $3827 + 12.54 + 1.567$ using four-decimal floating-point arithmetic

(i) starting at the left
(ii) starting at the right.

The exact value of the sum is 3841.107. Which result is the more accurate? Can you say (without detailed error analysis) why it is best to start the summation with the smallest pair of numbers?

Solution

Method (*i*) First, $3827 + 12.54$ is evaluated as

$$\begin{aligned}&\text{fl}\{10^4 \times 0.3827 + 10^2 \times 0.1254\}\\&\quad = \text{fl}\{10^4(0.3827 + 0.001\,254)\}\\&\quad = \text{fl}\{10^4 \times 0.383\,954\}\\&\quad = 10^4 \times 0.3840.\end{aligned}$$

Adding in the last term gives

$$\begin{aligned}\text{fl}\{10^4(0.3840 + 0.000\,156\,7)\} &= \text{fl}\{10^4 \times 0.384\,156\,7\}\\&= 10^4 \times 0.3842.\end{aligned}$$

Method (*ii*) First, $1.567 + 12.54$ is evaluated as

$$\begin{aligned}\text{fl}\{10^2(0.015\,67 + 0.1254)\} &= \text{fl}\{10^2 \times 0.141\,07\}\\&= 10^2 \times 0.1411.\end{aligned}$$

Adding in the last term gives

$$\begin{aligned}\text{fl}\{10^4(0.001\,411 + 0.3827)\} &= \text{fl}\{10^4 \times 0.384\,111\}\\&= 10^4 \times 0.3841.\end{aligned}$$

If ε_1 and ε_2 are bounds on the absolute errors introduced by the two additions, an absolute error bound for the result is $\varepsilon_1 + \varepsilon_2$. In the text it is stated that an error bound for a floating-point addition operation is 5×10^{-4} times the result. To minimize the absolute error introduced by the first addition, we should make its result as small as possible, i.e. perform $12.54 + 1.567$ not 3827 + something. (The error in the second addition cannot be altered much, since the sum is bound to be about 3841.)

7.1.5 An Elementary Case Study

To illustrate the various points made so far in this unit, let us consider the problem of evaluating the integral

$$\int_0^1 x^9 e^{(x-1)}\,dx.$$

Suppose, as is the habit of mathematicians, that we generalize this a bit and consider the integral

$$I_r = e^{-1}\int_0^1 x^r e^x\,dx,*$$

and enquire how we might compute I_r for $r = 0, 1, 2, \ldots$. There are at least four possible methods.

(i) *Finite series* Using the *Techniques of Integration* handbook, **TI**, we find that I_r can be expressed as the *finite* series

$$I_r = 1 - r + r(r-1) + \cdots + (-1)^{r-1}r! + (-1)^r r!(1 - e^{-1}),$$

(see Exercise 1, at the end of this sub-section).

This is in a sense a reformulation of the problem. Is it satisfactory for numerical purposes? We can certainly evaluate the series for any value of r, but how accurate will be the result computed by our hypothetical four-digit decimal machine?

Trying $r = 6$, we get

$$I_6 = 1 - 6 + 30 - 120 + 360 - 720 + 720(1 - e^{-1}).$$

Now e^{-1} is 0.367 879 ..., but our machine can store only the rounded version 0.3679, with an error of 0.000 020 .... The error in the computed I_6 is then 720 times this error, about 0.014, and since I_6 turns out to be about 0.127 we have a relative error of more than 10%. For I_9, for which the factor multiplying $(1 - e^{-1})$ is 362 880, the error from this source is as much as 7.2, many times greater than the true value of I_9, which is about 0.09.

For the computation of I_9 there is another source of error. The series is

$$1 - 9 + 72 - 504 + 3024 - 15\,120 + 60\,480 - 181\,440 + 362\,880 - 362\,880(1 - e^{-1}),$$

and the last three terms cannot be stored accurately in floating-point form.

Using the word "formulation" in a slightly more general sense than previously, it is clear that we have formulated an ill-conditioned mathematical problem (since small errors in the "data" e^{-1} are multiplied by large numbers to produce large errors in the answer), and that the method of solution leads to induced instability (since rounding errors in the calculation of such numbers as 362 880 are large and can seriously affect the computed result).

(ii) *Numerical integration* A possibility for avoiding the difficulties we encountered above in evaluating I_r is to compute it, for each r, by numerical integration, using, for example, Simpson's rule (see *Unit M100 9, Integration I*). The formula is

* The evaluation of this integral is the subject of a Computer Exercise contained in the supplementary booklet for this unit.

$$\int_0^1 f(x)\,dx = \tfrac{1}{3}h\{f_0 + 4f_1 + 2f_2 + 4f_3 + 2f_4 + \ldots + 2f_{2n-2} + 4f_{2n-1} + f_{2n}\}$$

where $2nh = 1$, and $f_k = f(kh)$ $(k = 0, 1, \ldots, 2n)$.

In our particular case this becomes

$$\int_0^1 x^r e^x\,dx = \tfrac{1}{3}h\{0 + 4(h^r)e^h + 2(2h)^r e^{2h} + \cdots + e\}$$

and in order to obtain I_r, we have to multiply this by e^{-1}. This is a better idea, the reformulated problem being well-conditioned (any "error" in e^{-1} clearly produces only the same relative error in the result), and the method has little induced instability. In fact, the integrand is positive everywhere and the addition of positive numbers cannot, through rounding errors, produce a large relative error; it was the *cancellation* of large numbers in method (i), producing a small result with an absolute error appropriate to a large result, which caused the induced instability.

On the other hand, we now have a lot of computation, including many evaluations of e^x. Moreover, Simpson's rule is not exact for this integrand, and we either have to estimate the error by some non-trivial analysis or, more practically, use a sequence of decreasing intervals h in the Simpson formula. As h approaches 0 the Simpson approximation approaches the true value of the integral, and we can estimate the accuracy by inspection. For example, with $h = \frac{1}{2}, \frac{1}{4}$ and $\frac{1}{8}$ the four-figure Simpson-rule estimates of I_6 are respectively 0.1730, 0.1312 and 0.1271, and $h = \frac{1}{16}$ gives 0.1268, and we have some confidence that this is correct to four figures. (The rate of convergence to the true result can be improved by a device called *Romberg integration*, which you will find described in *Computing Methods for Scientists and Engineers*, by Fox and Mayers (see Bibliography).

(iii) *A recurrence formula* In methods (i) and (ii) the computation of I_r was performed independently for each value of r; but when we want to calculate a number which can be regarded as a member of a sequence, in this case the sequence $I_0, I_1, I_2, \ldots$ it is often useful to look for a formula connecting each element to one or more of its immediate predecessors in the sequence. In *Unit M100 7, Sequences and Limits I*, we called these formulas recurrence formulas; but they are more commonly called *recurrence relations*. For the sequence $I_0, I_1, \ldots$, a recurrence relation can be found using integration by parts (see Exercise 1 of this sub-section), viz:

$$I_r = 1 - rI_{r-1} \qquad (r = 1, 2, 3, \ldots)$$

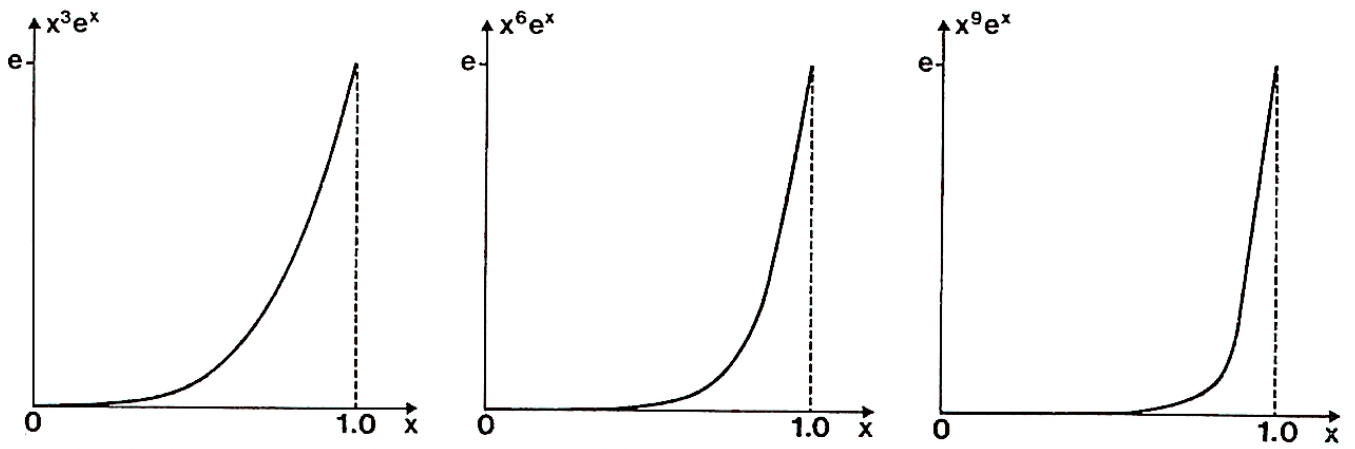
If we know one member of the sequence, say I_0, we can compute $I_1, I_2, \ldots$ in succession, from this recurrence relation. So here is another reformulation of the problem of calculating I_r, apparently very attractive and arithmetically simple. Let us try it out by performing the computation on a four-digit machine. With $I_0 = 1 - e^{-1} = 0.6321$ in our machine, we would find the results

r	0	1	2	3	4
I_r	0.6321	0.3679	0.2642	0.2074	0.1704

r	5	6	7	8
I_r	0.1480	0.1120	0.2160	−0.7280

Something is obviously wrong, since clearly I_r should always be positive. In fact, the poor results (in particular, for I_8) are due solely to a large magnification of the original error in I_0. Again we have an ill-conditioned formulation! In this particular calculation there is no induced instability because all the arithmetic happens to be exact. But if there had been any rounding errors in the successive applications of the recurrence relation these would also have been magnified, and the method would have shown induced instability as well as ill-conditioning.

(iv) *Backward recurrence* The recurrence relation is such an attractive idea that we do not abandon it immediately, but look for some other information which, while theoretically equivalent to what we have already used, might lead to a well-conditioned formulation of the problem. We don't want to compute directly any other value of I_r, but what we can do is consider what happens when r is large.



For large r, the factor x^r in the integrand, $x^r e^x$, is very small (except in a small region where x is very close to 1), and so we expect that I_r will be very small. (In fact the integrand lies between 0 and ex^r, and so eI_r lies between 0 and

$$\int_0^1 ex^r\,dx = \frac{e}{r+1},$$

which approaches 0 for large r.)

This suggests the possibility of doing the recurrence in the reverse direction, using the recurrence relation in the form

$$I_{r-1} = \frac{1}{r}(1 - I_r)$$

and starting with the approximation of taking $I_n = 0$ for some large n.

Trying it out for $n = 10$, 12 and 14, we would obtain, using our hypothetical four-figure machine

r	$I_r(n = 10)$	$I_r(n = 12)$	$I_r(n = 14)$
14			0
13			0.07143
12		0	0.07143
11		0.08333	0.07738
10	0	0.08334	0.08387
9	0.1000	0.09167	0.09161
8	0.1000	0.1009	0.1009
7	0.1125	0.1124	0.1124
6	0.1268	0.1268	0.1268
5	0.1455	0.1455	0.1455
4	0.1709	0.1709	0.1709
3	0.2073	0.2073	0.2073
2	0.2642	0.2642	0.2642
1	0.3679	0.3679	0.3679
0	0.6321	0.6321	0.6321

The results indicate that at last we have found a well-conditioned mathematical formulation of the problem, at least for all r with $r \leqslant 8$. This time our "data" is the value of n, and we see that as far as the calculation of I_8 is concerned it does not matter whether we take $n = 12$ or $n = 14$ or (presumably) some even better "approximation" to the "correct value" of n (i.e. some even larger value)—we always get $I_8 = 0.1009$ to 4 figures. Moreover, the method has no induced instability; although rounding errors do occur at every step when we divide by r, they do not affect the calculated value of I_r for $r \leqslant 8$.

This case study shows, in the first place, how important it is when solving a problem to investigate methods closely, to correct when possible both inherent and induced instability. It also illustrates how recurrence relations, if handled correctly, can give very convenient and powerful methods in numerical computation; but that, if handled incorrectly, they can give nonsense. In order to be able to use them effectively, therefore, you need to know something about the theory of recurrence relations and how to analyse the errors in numerical calculations based on them. In the remainder of this unit we will discuss each of these points further.

Exercises

1. Use the method of integration by parts to obtain the formula

$$I_r = 1 - r + r(r-1) + \cdots + (-1)^{r-1} r! + (-1)^r r!(1 - e^{-1}),$$

where $I_r = e^{-1} \int_0^1 x^r e^x \, dx$.

2. Consider the problem of evaluating e^{-10}, which is about 0.000 05. The obvious way is to substitute -10 for x in the Maclaurin series

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$$

obtaining

$$e^{-10} = 1 - 10 + \frac{100}{2} - \frac{1000}{6} + \frac{10\,000}{24} - \frac{100\,000}{120} + \frac{1\,000\,000}{720} - \cdots$$

The series is convergent, but it is not a good method of calculating e^{-10}, because of induced instability.

(i) What feature of the series should put you on the look-out for induced instability?

(ii) What accuracy would you expect in your value for e^{-10} if you used this method to calculate e^{-10} on the hypothetical four-digit computer?

(iii) Suggest a better method for calculating e^{-10}, giving an accuracy of nearly 4 significant figures. (*Hint* Look at the method suggested in Example 1 of sub-section 7.1.2, for avoiding induced instability in the solution of a quadratic equation.)

Solutions

1. $$I_r = e^{-1}\int_0^1 x^r e^x\,dx$$
$$= e^{-1}\int_0^1 x^r \frac{d}{dx}(e^x)\,dx$$
$$= e^{-1}\,[x^r e^x]_0^1 - e^{-1}\int_0^1 r x^{r-1} e^x\,dx,$$

by integration by parts.

Thus,

$$I_r = 1 - rI_{r-1} \qquad (r = 1, 2, 3, \ldots),$$

from which

$$I_{r-1} = 1 - (r-1)I_{r-2}$$

and hence

$$I_r = 1 - r(1 - (r-1)I_{r-2})$$
$$= 1 - r + r(r-1)I_{r-2}$$

This process can be repeated, giving

$$I_r = 1 - r + r(r-1) - r(r-1)(r-2)I_{r-3}$$

and

$$I_r = 1 - r + r(r-1) - r(r-1)(r-2) + r(r-1)(r-2)(r-3)I_{r-4}$$

The final stage yields

$$I_r = 1 - r + r(r-1) + \cdots + (-1)^{r-1} r!\,(1 - (r - (r-1))I_0)$$
$$= 1 - r + r(r-1) + \cdots + (-1)^{r-1} r! + (-1)^r r!(1 - e^{-1}),$$

since

$$I_0 = 1 - e^{-1}.$$

2. (i) The calculation involves the subtraction of large numbers to produce a result which is relatively small (between 0 and 1).

(ii) No accuracy at all can be expected; that is to say, the number obtained might bear no relation to the true value of e^{-10}. The term 10 000/24, for example, would be stored in the computer as 0.4167×10^3, which is in error by at least $0.000\,03 \times 10^3 = 0.03$. This error alone is already much bigger than the sum we want to calculate.

(iii) A better method would be to calculate e^{10} from the series $e^{10} = 1 + 10 + \frac{100}{2!} + \frac{1000}{3!} + \cdots$, and then calculate its reciprocal. This time, all the terms of the series are positive, so that there is no cancellation and no induced instability in its evaluation.

7.1.6 Summary of Section 7.1

In this section we defined the terms

ill-conditioned problem	(page C 7)	* * *
well-conditioned problem	(page C 7)	* * *
inherent instability	(page C 7)	* * *
magnification factor	(page C 7)	* *
absolute condition number	(page C 7)	* *
relative condition number	(page C 8)	* *
induced instability	(page C 10)	* * *
floating-point arithmetic	(page C 12)	* * *
relative error	(page C 12)	* * *

Techniques

1. Use condition numbers to predict ill-conditioning. * *
2. Perform calculations in four-figure floating-point arithmetic and analyse the errors involved. * * *
3. Consider various methods for a given problem choosing, if possible, the most stable and most economic one (as exemplified in the case study). * * *

Notation

ε	(page C 6)
x	(page C 6)
X	(page C 6)
$\mathrm{fl}(x)$	(page C 12)
r_x	(page C 12)
r	(page C 14)

7.2 RECURRENCE RELATIONS

7.2.1 What is a Recurrence Relation?

The formula we used in the last section,

$$I_r + rI_{r-1} = 1 \qquad (r = 1, 2, \ldots)$$

is an example of a recurrence formula or recurrence relation. In general, if we have any sequence of numbers, say $u_0, u_1, u_2, \ldots$, then any formula connecting a general element u_r of the sequence with its predecessors in the sequence is called a *recurrence relation*. A particularly simple form of recurrence relation is

$$u_r = a_r u_{r-1} + b_r \qquad (r = 1, 2, \ldots)$$

where a_r and b_r may depend on r. This is called a *linear, first-order* recurrence relation (we shall see why in sub-section 7.2.3). The recurrence relation $I_r + rI_{r-1} = 1$ is evidently of this type (since it can be written $I_r = -rI_{r-1} + 1$); so are the recurrence relations involved in arithmetic and geometric series:

Arithmetic series $a, a + h, a + 2h, \ldots$ $(u_r = u_{r-1} + h)$

Geometric series $a, ax, ax^2, \ldots$ $(u_r = xu_{r-1})$

Another example of a recurrence relation is the one defining the Fibonacci sequence, which we discussed in *Unit 5, Determinants and Eigenvalues*; in the notation of this section it is

$$u_r = u_{r-1} + u_{r-2} \qquad (u_0 = u_1 = 1;\ r = 2, 3, \ldots)$$

This is an example of a linear second-order recurrence relation. In general, any recurrence formula that can be put in the form $u_r = g(r, u_{r-1}, u_{r-2})$, but not $u_r = f(r, u_{r-1})$, is said to be of *second order*.

The following exercises illustrate some of the many ways that recurrence relations can arise in practice.

Exercises

1. (i) If $J_r = \int_0^{\pi/2} x^r \cos x \, dx \qquad (r = 0, 1, \ldots)$,

show (by integrating by parts twice) that J_r satisfies the linear second-order recurrence formula

$$J_r = \left(\frac{\pi}{2}\right)^r - r(r-1)J_{r-2} \qquad (r = 2, 3, \ldots).$$

(ii) Given that $J_0 = 1$, write down the exact values of J_2 and J_4.

2. In the Foundation Course you met the definition of a derivative

$$y'(x) = \lim_{h \to 0} \frac{y(x+h) - y(x)}{h}.$$

We can approximate $y'(x)$ by the formula $\dfrac{y(x+h) - y(x)}{h}$ for some small (but not zero) value of h (if $y'(x)$ exists).

(i) Write down the recurrence relation obtained by using this approximation in the differential equation

$$y'(x) = x^2 + y(x)$$

and by replacing x by rh. (Use the notation

$$u_r = y(rh) \ (r = 0, 1, \ldots),$$

so that $y'(rh) \simeq (u_{r+1} - u_r)/h$.)

(ii) Given that $y(0) = 1$, and using $h = 0.1$, calculate approximations to $y(0.1)$ and $y(0.2)$.

Because of this type of application, recurrence relations are often called *difference equations*.

If you have time, you might like to solve the differential equation by the method described in *Unit 4, Differential Equations I* and compare the results.

Solutions

1. (i) $J_r = \int_0^{\pi/2} x^r \cos x \, dx \qquad (r = 0, 1, 2, \ldots)$

$$= \int_0^{\pi/2} x^r \frac{d}{dx}(\sin x)\, dx$$

$$= [x^r \sin x]_0^{\pi/2} - \int_0^{\pi/2} r x^{r-1} \sin x \, dx$$

$$= \left(\frac{\pi}{2}\right)^r - \int_0^{\pi/2} r x^{r-1} \frac{d}{dx}(-\cos x)\, dx$$

$$= \left(\frac{\pi}{2}\right)^r + [r x^{r-1} \cos x]_0^{\pi/2} - \int_0^{\pi/2} r(r-1) x^{r-2} \cos x \, dx$$

$$= \left(\frac{\pi}{2}\right)^r + 0 - \int_0^{\pi/2} r(r-1) x^{r-2} \cos x \, dx$$

i.e. $J_r = \left(\frac{\pi}{2}\right)^r - r(r-1) J_{r-2} \qquad (r = 2, 3, \ldots)$

(ii) To find J_2 we set $r = 2$, obtaining

$$J_2 = (\pi/2)^2 - 2 \times (2-1) \times 1 \quad \text{since} \quad J_0 = 1$$
$$= (\pi/2)^2 - 2.$$

To find J_4 we set $r = 4$, obtaining

$$J_4 = (\pi/2)^4 - 4 \times 3 \times [(\pi/2)^2 - 2]$$
$$= (\pi/2)^4 - 12(\pi/2)^2 + 24.$$

2. (i) The approximation suggested, applied to

$$y'(x) = x^2 + y(x)$$

gives

$$\frac{u_{r+1} - u_r}{h} = (rh)^2 + u_r$$

or

$$u_{r+1} = (1 + h)u_r + r^2 h^3.$$

This is the required recurrence relation (it gives u_{r+1} in terms of u_r rather than u_r in terms of u_{r-1} as we have had up to now; but this is a difference of notation, not of substance.)

(ii) With $h = 0.1$, the recurrence formula becomes

$$u_{r+1} = (1.1)u_r + 10^{-3} \times r^2$$

To find $y(0.1)$, which is u_1, we set $r = 0$, obtaining

$$y(0.1) = u_1 = (1.1)u_0 = 1.1$$

since $u_0 = y(0) = 1$ (given).

To find $y(0.2)$ we set $r = 1$ and get

$$y(0.2) = u_2 = (1.1)u_1 + 10^{-3} \times 1^2$$
$$= 1.21 + 0.001$$
$$= 1.211.$$

The differential equation

$$y'(x) - y(x) = x^2$$

can be solved by using an integrating factor.

Thus

$$\frac{d}{dx}(y(x) \exp(-x)) = x^2 \exp(-x)$$

i.e.

$$y(x) \exp(-x) = \int x^2 \exp(-x)\, dx.$$

Using integration by parts twice

$$y(x) = -x^2 - 2x - 2 + 3 \exp(x)$$

since

$$y(0) = 1.$$

Hence

$$y(0.1) = 1.106,$$

and

$$y(0.2) = 1.224.$$

7.2.2 Linear Recurrence Relations of First Order

The easiest recurrence relations to treat theoretically are those of the first order. In this sub-section we will obtain the theoretically exact solution of any first-order linear recurrence relation (by "theoretically," we mean that the solution may not be in a form suitable for practical calculations).

We consider a special case first: the recurrence relation

$$u_r = au_{r-1} + b \qquad (r = 1, 2, \ldots)$$

where a and b are fixed real numbers (i.e. independent of r). Applying this in turn with $r = 1$, $r = 2$ and so on we can express u_1, u_2, u_3, etc. in terms of u_0. The recurrence relation itself does not tell us the value of u_0; we shall treat it as an arbitrary constant, analogous to the arbitrary constant in the solution of a first-order differential equation. Denoting this arbitrary constant by c and applying the recurrence relation we obtain

$$\begin{aligned} u_0 &= c \\ u_1 &= au_0 + b = ac + b \\ u_2 &= au_1 + b = a^2c + (a+1)b \\ u_3 &= au_2 + b = a^3c + (a^2 + a + 1)b \end{aligned}$$

and in general

$$u_r = a^r c + (a^{r-1} + a^{r-2} + \cdots + a + 1)b \quad (r = 1, 2, \ldots).$$

Using the formula for the sum of a geometric progression this can be written in one of two alternative forms:

$$\left.\begin{array}{ll} \text{if } a \neq 1, & u_r = a^r c + \dfrac{a^r - 1}{a - 1} b \qquad (r = 1, 2, \ldots) \\ \text{if } a = 1, & u_r = c + rb \qquad\qquad\quad (r = 1, 2, \ldots) \end{array}\right\} \tag{1}$$

We can generalize this calculation to the case where a and b depend on r, so that the recurrence relation has the form

$$u_r = a_r u_{r-1} + b_r \qquad (r = 1, 2, \ldots)$$

Again taking u_0 to be an arbitrary constant and calling it c, and then using the recurrence relation with $r = 1$, then $r = 2$, then $r = 3$, and so on, we find

$$\begin{aligned} u_0 &= c \\ u_1 &= a_1 u_0 + b_1 = a_1 c + b_1 \\ u_2 &= a_2 u_1 + b_2 = a_2 a_1 c + a_2 b_1 + b_2 \\ u_3 &= a_3 u_2 + b_3 = a_3 a_2 a_1 c + a_3 a_2 b_1 + a_3 b_2 + b_3 \end{aligned}$$

and in general

$$\begin{aligned} u_r = (a_r a_{r-1} \ldots a_2 a_1)c + (a_r a_{r-1} \ldots a_3 a_2)b_1 + (a_r a_{r-1} \ldots a_4 a_3)b_2 \\ + \cdots + a_r b_{r-1} + b_r \qquad (r = 1, 2, \ldots). \end{aligned}$$

This can be written more concisely using the summation sign:

$$u_r = (a_r a_{r-1} \cdots a_2 a_1)c + \sum_{s=1}^{r} (a_r a_{r-1} \cdots a_{s+1})b_s \tag{2}$$

with the convention that for $s = r$ in the summation, the product $a_r \cdots a_{s+1}$ is 1. This formula is analogous to the formula at the top of page K97 which gives the general solution of a first-order linear differential equation. Notice that it gives u_r as the sum of two parts, one proportional to an arbitrary constant c and the other independent of c. We shall see later in the unit that these two parts correspond to the two parts in the solution of a linear problem: the first is in the kernel of a linear transformation associa-

ted with the recurrence relation, and the second is a particular solution of the recurrence relation.

Example 1

The general solution of the recurrence relation

$$u_r = 1 - ru_{r-1} \qquad (r = 1, 2, \ldots)$$

which we used in our case study (sub-section 7.1.5), is obtained by setting $a_r = -r$ and $b_r = 1$ in the general linear recurrence relation $u_r = a_r u_{r-1} + b_r$. Thus, using the formula of Equation 2, we obtain

$$\begin{aligned} u_r &= (-r)(-(r-1)) \cdots (-2)(-1)c \\ &\quad + \sum_{s=1}^{r} (-r)(-(r-1)) \cdots (-(s+1))1 \\ &= (-1)^r r!c + \sum_{s=1}^{r} (-1)^{r-s} r(r-1) \cdots (s+1), \end{aligned}$$

from which

$$\begin{aligned} u_0 &= c \\ u_1 &= -c + 1 \\ u_2 &= 2c + (-1) \times 2 + (-1)^0 \times 1 = 2c - 1 \\ u_3 &= -6c + (-1)^2 \times 3 \times 2 + (-1)^1 \times 3 + (-1)^0 \times 1 = -6c + 4. \end{aligned}$$

Any value of c gives a solution of the recurrence relation. If we want u_r to be equal to I_r, which is $e^{-1}\int_0^1 x^r e^x \, dx$, we need an extra condition to determine c. Theoretically, the obvious choice is the known value of I_0, which is $e^{-1}\int_0^1 e^x dx = 1 - e^{-1}$. Thus, since $c = I_0$, $c = 1 - e^{-1}$; with this value of c, the above solution gives

$$I_r = (-1)^r r!(1 - e^{-1}) + \sum_{s=1}^{r} (-1)^{r-s} r(r-1) \cdots (s+1).$$

Practically, however, as we have seen in the case study, this formula is not suitable for the numerical evaluation of I_r unless r is quite small, since e^{-1} is not known exactly and is multiplied by a large factor.

Exercises

1. Write down the general solutions of the following recurrence relations, by using the formulas in Equations (1) and (2):

   (i) $u_r = 2u_{r-1}$
   (ii) $u_r - ru_{r-1} = 0$
   (iii) $u_{r+1} - (r+1)u_r = 0$

2. For the recurrence relations in the previous question, pick out the particular solutions that satisfy $u_0 = 1$.
3. You borrow £S from a building society on 1st January at $I\%$ per annum interest and pay back £P every 1st January thereafter. If the amount you owe them just after your rth repayment is £u_r, find a recurrence relation connecting u_r with u_{r-1} by filling in the blanks below.

   Just after the rth repayment, you owe

   _____

   Just before the $(r+1)$th repayment, you owe

   _____

Just after the $(r+1)$th repayment, you owe

_____

Hence $u_{r+1} =$ _____

Hence find a formula for the amount you owe after the rth repayment, in terms of r, S, P and I.

Solutions

1. (i) Equation (1) with $a = 2$, $b = 0$, gives $u_r = 2^r c$.
   (ii) Equation (2) with $a_r = r$, $b_r = 0$, gives $u_r = r!c$.
   (iii) This is identical with the previous case. All that has happened is that we have $r+1$ where r was before; so the answer is

$$u_{r+1} = (r+1)!c,$$

   which is equivalent to

$$u_r = r!c.$$

2. Since $u_0 = c$, the appropriate choices from the general solutions given above are (i) $u_r = 2^r$; (ii) and (iii) $u_r = r!$.
3. Just after the rth repayment, you owe £u_r. Just before the $(r+1)$th repayment, you owe

$$£u_r + \frac{I}{100}£u_r = £\left(1 + \frac{I}{100}\right)u_r.$$

   Just after the $(r+1)$th repayment, you owe

$$£\left(1 + \frac{I}{100}\right)u_r - £P.$$

   Hence

$$u_{r+1} = \left(1 + \frac{I}{100}\right)u_r - P \qquad (r = 0, 1, \ldots)$$

   This is of the form $u_r = au_{r-1} + b$ with $a = 1 + \frac{I}{100}$, $b = -P$, and with r replaced by $r+1$.

   The general solution of the recurrence relation is therefore the one given in Equation (1),

$$u_r = a^r c + \frac{a^r - 1}{a - 1} b$$

$$= \left(1 + \frac{I}{100}\right)^r c + \frac{\left(1 + \frac{I}{100}\right)^r - 1}{\left(1 + \frac{I}{100}\right) - 1} \times (-P)$$

   To find the solution appropriate to our case we use the one remaining piece of information, which is that the amount owed initially (i.e. exactly one year before the first repayment) is £S. In our notation, this is £u_0, and therefore we have $c = S$ and so the answer to the problem is

$$u_r = \left(1 + \frac{I}{100}\right)^r S - \frac{100P}{I}\left\{\left(1 + \frac{I}{100}\right)^r - 1\right\}.$$

7.2.3 Linear Recurrence Relations as Linear Problems

In sub-section 7.2.1 we promised to explain why we are entitled to call the recurrence relation

$$u_r = a_r u_{r-1} + b_r$$

linear. Actually, such an explanation is not entirely straightforward because the word "linear" has a very special meaning in this course and the use of the word in this context, whilst fully justified, is one step removed from our usage. In the course we use it to describe a special type of mapping between vector spaces. So where are the vector spaces here?

The vector space is the same in the domain and the codomain. It is the space of all infinite sequences of real numbers: an element of the space has the form

$$\mathbf{u} = (u_0, u_1, u_2, \ldots)$$

where the u_r are real numbers and the list of numbers goes on for ever. It is an obvious extension of the concept of the spaces

R^2 (pairs of numbers)

R^3 (triples of numbers)

.....

R^n (n-tuples of numbers, where n is any fixed positive integer)

and the operations are very similar.

$$(u_0, u_1, u_2, \ldots) + (v_0, v_1, v_2, \ldots) = (w_0, w_1, w_2, \ldots)$$

where $w_r = u_r + v_r$, and

$$\alpha(u_0, u_1, u_2, \ldots) = (w_0, w_1, w_2, \ldots)$$

where $w_r = \alpha u_r$. A useful symbol for the space is R^∞.

The problem of solving a recurrence relation such as

$$u_r = a_r u_{r-1} + b_r$$

is equivalent to finding a sequence $\mathbf{u}$, such that

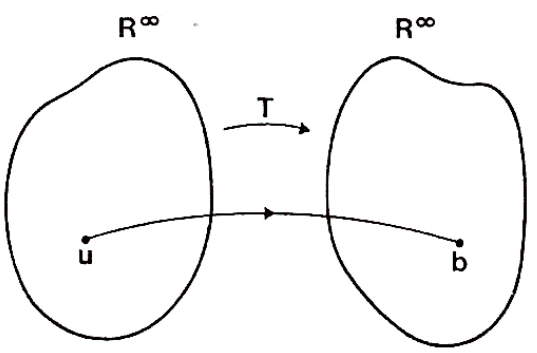
$$u_r - a_r u_{r-1} = b_r.$$

This problem fits the format of our general linear problem, for putting $u_r - a_r u_{r-1} = w_r (r = 1, 2, \ldots)$, the general term of a sequence $\mathbf{w}$, and $w_0 = 0$, we define a mapping

$$T: \mathbf{u} \longmapsto \mathbf{w}.$$

Thus, in trying to solve the recurrence relation, we are trying to solve the equation

$$T(\mathbf{u}) = \mathbf{b}$$



It remains to show that T is a linear transformation, in which case the recurrence relation we started with simply specifies a linear problem and so it would seem reasonable to describe it as a *linear* recurrence relation.

Exercise

Show that T is a linear transformation.

Solution

$$T(\mathbf{u} + \mathbf{v}) = \mathbf{w},$$

where

$$\begin{aligned} w_r &= u_r + v_r - a_r(u_{r-1} + v_{r-1}) \\ &= u_r - a_r u_{r-1} + v_r - a_r v_{r-1} \end{aligned}$$

i.e.

$$T(\mathbf{u} + \mathbf{v}) = T(\mathbf{u}) + T(\mathbf{v})$$

Also

$$T(\alpha\mathbf{u}) = \mathbf{w}$$

where

$$\begin{aligned} w_r &= \alpha u_r - a_r \alpha u_{r-1} \\ &= \alpha(u_r - a_r u_{r-1}) \end{aligned}$$

i.e.

$$T(\alpha\mathbf{u}) = \alpha T(\mathbf{u}).$$

Thus T is a linear transformation.

There is a whole class of linear transformations of R^∞ to itself which bear a very close relationship to the linear differential operators that we met in *Unit 4, Differential Equations I*. One obvious similarity is that in both cases the vector spaces have infinite dimension. But there is another similarity. Linear differential operators are built up from a simple constituent linear transformation, D, the differentiation operator. The same is true in the case of linear transformations from R^∞ to itself which lead to recurrence relations; they are built up from the transformation E defined by

$$E: (u_0, u_1, u_2, \ldots) \longmapsto (u_1, u_2, u_3, \ldots).$$

You might call it the "beheading mapping": its effect is to shift every term in the sequence up one place nearer the head, and to lose the "head", u_0. E is commonly called the *shift operator*.

Exercise

Show that E is a linear transformation.

Solution

$$\begin{aligned} E(u_0 + v_0, u_1 + v_1, \ldots) &= (u_1 + v_1, u_2 + v_2 \ldots) \\ &= (u_1, u_2, \ldots) + (v_1, v_2, \ldots) \\ &= E(u_0, u_1, u_2, \ldots) \\ &\quad + E(v_0, v_1, v_2, \ldots) \\ E(\alpha u_0, \alpha u_1, \alpha u_2, \ldots) &= (\alpha u_1, \alpha u_2, \ldots) \\ &= \alpha(u_1, u_2, \ldots) \\ &= \alpha E(u_0, u_1, u_2, \ldots) \end{aligned}$$

In *Unit 4, Differential Equations I* we saw how D can be used to build up more complicated linear differential operators. Much the same thing can be done with E. For example,

$$E + 2 : \mathbf{u} \longmapsto E\mathbf{u} + 2\mathbf{u},$$

$$E^2 + 3E + 1 : \mathbf{u} \longmapsto E(E\mathbf{u}) + 3E(\mathbf{u}) + \mathbf{u},$$

$$(E + 1)(E + 2) : \mathbf{u} \longmapsto (E + 1)((E + 2)\mathbf{u})$$

are all *linear difference operators*.

The comparison does not end here—the methods of solution in each case are strikingly similar. There is, of course, an obvious similarity arising from the fact that they are both linear problems, but also the actual *technique* of finding the kernel is much the same. This is hardly surprizing because we know that E has something to do with differentiation; in the Foundation Course we introduced differentiation by way of differencing. But there is also a similarity between the vector spaces, apart from the fact that they are both dimensionally infinite. The domain and codomain of D consist of continuous functions. The space R^∞ could also be identified with a space of functions having domain Z_0^+; for a sequence $\mathbf{u}$ can be identified with a function

$$f: r \longmapsto u_r \qquad (r \in Z_0^+).$$

The vector space operations in R^∞ then correspond to ordinary addition of functions and multiplication of functions by real numbers.

As with differential equations, we classify recurrence relations by their *order*. The order of a recurrence relation is simply the highest power of E in the equation. Thus

$$(E + 3)\mathbf{u} = 0$$

is *first order*,

$$(E^2 + 4E - 3)\mathbf{u} = 0$$

is *second order*, and so on. When the equation is written in terms of the coordinates of $\mathbf{u}$, such as

$$u_r + 3u_{r-1} = 2,$$

the order is simply the difference between the largest and smallest suffices —in this example the order is 1. The equation

$$u_r + ru_{r-1} + 4u_{r-2} = 3^r$$

is *second order*, and so on.

Solving recurrence relations

As with differential equations, the first step in solving a recurrence relation is to find the kernel. The analogy continues because, as with differential equations, the dimension of the kernel is the same as the order of the recurrence relation (see *Unit 4*, sub-section 4.2.2). For example, in the first-order case, the kernel will be the solution of a problem such as

$$u_{r+1} - a_r u_r = 0$$

or

$$u_{r+1} = a_r u_r.$$

We wish to show that the solution space of this problem is one-dimensional and we can do this by showing that it is isomorphic to the one-dimensional space R. Given any $u_0 \in R$, the sequence $\mathbf{u}$ is defined uniquely by

$$u_1 = a_0 u_0$$

$$u_2 = a_1 a_0 u_0$$

and so on. The mapping

$$f_0 : \mathbf{u} \longmapsto u_0$$

from the solution space to R is one–one. It is easy to complete the proof that f_0 is an isomorphism (notice that u_0 can be *any* element of R), and so the kernel of the first-order problem is one-dimensional. The single sequence

$$(a_0, a_1 a_0, a_2 a_1 a_0, \ldots)$$

is a basis for it.

A similar analysis can be adopted for the more general cases. Thus to find the kernel for an nth-order recurrence relation, we need to find n linearly independent solution vectors to form a basis. The way these are found is again very similar to the way the problem is tackled in differential equations. To illustrate the technique, and because we shall be needing the results later in the unit, we shall have a look at second-order, constant-coefficient recurrence relations, i.e. relations of the form

$$(E^2 + aE + b)\mathbf{u} = \mathbf{w}$$

where a and b are real numbers. The kernel of the linear transformation $E^2 + aE + b$ is the solution set of the homogeneous equation

$$(E^2 + aE + b)\mathbf{u} = \mathbf{0}.$$

In the analogous case with differential equations, the kernel had as a basis functions of the form $x \longmapsto e^{\lambda x}$ (see sub-section 2.2 of *Unit 4, Differential Equations I*). This gives us a clue as to how to proceed in this case if we notice an interesting feature of these functions. They satisfy the equation

$$Df = \lambda f.$$

That is to say, the basis of the kernel consists of eigenvectors of D. (Notice that D is a linear transformation from an infinite-dimensional space and has an infinite number of eigenvalues: λ can be *any* real number. To find a basis for the kernel we select just the ones we need.)

The eigenvectors of E satisfy the equation

$$E\mathbf{u} = \lambda\mathbf{u}$$

i.e.

$$u_{r+1} = \lambda u_r$$

i.e.

$$u_{r+1} = \lambda^{r+1} u_0$$

for any $u_0 \in R$.

So the sequence $(1, \lambda, \lambda^2, \ldots)$ is an eigenvector of E for any value of λ. If we try such sequences as solutions of the homogeneous equation

$$(E^2 + aE + b)\mathbf{u} = \mathbf{0},$$

we find that λ must satisfy the equation

$$\lambda^2 + a\lambda + b = 0.$$

We refer to this as the *auxiliary equation* as in the other context.

Thus when this equation has two distinct roots, λ_1 and λ_2, the problem is solved; for the sequences

$$\lambda_1 = (1, \lambda_1, \lambda_1^2, \lambda_1^3, \ldots)$$
$$\lambda_2 = (1, \lambda_2, \lambda_2^2, \lambda_2^3, \ldots)$$

give two independent vectors which span the kernel. Thus for the problem

$$(E^2 + 3E + 2)\mathbf{u} = \mathbf{0}.$$

for example, we choose λ to take the values of the roots of

$$\lambda^2 + 3\lambda + 2 = 0$$

and so the solution space is

$$\langle (1, (-1), (-1)^2, (-1)^3, \ldots) \ , \ (1, (-2), (-2)^2, (-2)^3, \ldots) \rangle$$

i.e.

$$\langle \mathbf{u}, \mathbf{v} \rangle, \text{ where } u_r = (-1)^r \text{ and } v_r = (-2)^r, \ r = 0, 1, 2, \ldots.$$

Having found the kernel, we now need a particular solution of the non-homogeneous equation

$$(E^2 + aE + b)\mathbf{u} = \mathbf{w}$$

to complete the solution. We shall, however, not go into methods of finding the particular solution (see the example and exercises below), nor into the procedures to adopt when the λ's are equal or complex. If you are interested, you might like to try to continue the investigation when we tackle similar problems later for differential equations.

In sub-section 7.2.2, we have seen that it is a relatively straightforward matter to write down the solution set of first-order linear recurrence relations and some second-order ones. But even then we may be left with problems; for, whenever we want to calculate numerical values for these answers, i.e. *calculate* a specific u_r, we may find the answer swamped by errors. The numerical aspect is the main interest of this unit and so we shall turn to this problem in the next section.

Example

Find

(i) the general solution of the recurrence relation $u_r = u_{r-1} + u_{r-2}$;
(ii) the particular solution with $u_0 = u_1 = 1$ (in which case u_r is the rth Fibonacci number);
(iii) the general solution of $u_r = u_{r-1} + u_{r-2} + 1$.

(i) In this case the linear problem is homogeneous, and the solution is

$$u_r = c_1\lambda_1^r + c_2\lambda_2^r$$

where λ_1 and λ_2 are the roots of

$$\lambda^2 - \lambda - 1 = 0;$$

i.e. $\lambda_1 = \frac{1}{2}(1 - \sqrt{5})$ and $\lambda_2 = \frac{1}{2}(1 + \sqrt{5})$.

(ii) If $u_0 = u_1 = 1$, we have

$$u_0 = c_1 + c_2 = 1$$

$$u_1 = c_1\left(\frac{1-\sqrt{5}}{2}\right) + c_2\left(\frac{1+\sqrt{5}}{2}\right) = 1,$$

and the solution gives the rth Fibonacci number

$$u_r = \frac{1}{\sqrt{5}}\left\{\left(\frac{1+\sqrt{5}}{2}\right)^{r+1} - \left(\frac{1-\sqrt{5}}{2}\right)^{r+1}\right\}.$$

(This we found by a different method in *Unit 5, Determinants and Eigenvalues.*)

(iii) A particular solution, p_r, can be found by putting $p_r = \text{constant}$, since the non-homogeneous term in the recurrence relation is a constant. Putting $p_r = c_3 = \text{constant}$, we find $c_3 = c_3 + c_3 + 1$, giving $c_3 = -1$ and the general solution

$$u_r = c_1\left(\frac{1-\sqrt{5}}{2}\right)^r + c_2\left(\frac{1+\sqrt{5}}{2}\right)^r - 1.$$

Exercises

1. Write down the general solutions of
   (i) $u_r = u_{r-1} + 2u_{r-2}$
   (ii) $u_r + 5u_{r-1} + 6u_{r-2} = 0$
2. Solve $u_{r+2} + u_{r+1} = 6u_r$ with $u_0 = 0$, $u_1 = 1$.

3. Find the general solution of

$$u_{r+1} - 10.1u_r + u_{r-1} = -2.7r,$$

given that there is a particular solution of the form

$$u_r = c_3 r + c_4 .$$

Solutions

1. The auxiliary equations are

   (i) $\lambda^2 - \lambda - 2 = 0$ with roots $\lambda_1 = 2$, $\lambda_2 = -1$,
   (ii) $\lambda^2 + 5\lambda + 6 = 0$ with roots $\lambda_1 = -2$, $\lambda_2 = -3$.

   The general solutions are therefore

   (i) $u_r = c_1 2^r + c_2(-1)^r$,
   (ii) $u_r = c_1(-2)^r + c_2(-3)^r$.

2. Using the method of Solution 1 the general solution is $u_r = c_1 2^r + c_2(-3)^r$. The conditions $u_0 = 0$, $u_1 = 1$ give

$$\begin{aligned} c_1 + c_2 &= 0 \\ 2c_1 - 3c_2 &= 1; \end{aligned}$$

   so that $c_1 = \frac{1}{5}$, $c_2 = -\frac{1}{5}$ and the required solution is $u_r = \frac{1}{5}(2^r - (-3)^r)$.

3. The solution of the associated homogeneous problem is $c_1 10^r + c_2(\frac{1}{10})^r$, since 10 and $\frac{1}{10}$ are the roots of

$$\lambda^2 - 10.1\lambda + 1 = 0.$$

   If $c_3 r + c_4$ is a solution of the non-homogeneous equation, substitution gives

$$\begin{aligned} &(c_3(r+1) + c_4) - 10.1(c_3 r + c_4) + (c_3(r-1) + c_4) \\ &\quad = -2.7r, \end{aligned}$$

   which is true for all r if $c_3 = \frac{1}{3}$, $c_4 = 0$. The required solution is then

$$u_r = c_1 10^r + c_2(\tfrac{1}{10})^r + \tfrac{1}{3}r.$$

7.2.4 Summary of Section 7.2

In this section we defined the terms

recurrence relation	(page C 22)	* * *
linear recurrence relation	(page C 22)	* * *
linear difference operator	(page C 29)	* *
order of a recurrence relation	(page C 30)	* * *

Techniques

Solve first-order linear recurrence relations, and second-order linear recurrence relations with constant coefficients. * * *

Notation

R^∞	(page C 28)
E	(page C 29)

7.3 NUMERICAL ANALYSIS FOR RECURRENCE RELATIONS

7.3.1 First-order Recurrence Relations—Forward Recurrence

Even though we may be able to write down the general solution of a recurrence relation, there remains the problem of calculating the values of the numbers $u_0, u_1, \ldots$, of investigating whether small changes in the data make large or small changes in the members of the sequence (the question of inherent instability, ill-conditioning), and whether the rounding errors of computer arithmetic have serious or tolerable accumulation (the question of induced instability). In this sub-section, we do this for the problem of calculating $u_0, u_1, \ldots$ by "forward recurrence", that is, by starting with a given u_0 and then calculating u_1, u_2, and so on.

We first examine the question of inherent instability for the general first-order recurrence relation

$$u_r = a_r u_{r-1} + b_r \tag{1}$$

For simplicity we assume that all the numbers a_r and b_r are known exactly (and can be stored exactly), but that the initial value u_0 is inaccurate, either "physically" or "mathematically". We have seen (sub-section 7.2.2) that the general solution of Equation (1), corresponding to the initial value $u_0 = c$, is

$$u_r = (a_r \cdots a_1)c + \sum_{s=1}^{r} (a_r \cdots a_{s+1})b_s$$

Notice that, although we did not develop the solution in this way, the formula for u_r falls into two parts; the first part is an element of the kernel of the linear transformation

$$T : \mathbf{u} \longmapsto \mathbf{w},$$

where $w_r = u_r - a_r u_{r-1}$, and the second part is a particular solution of

$$T(\mathbf{u}) = \mathbf{b}.$$

For the purposes of calculation, since the as and bs are known exactly, we can regard each u_r as the image under some function $c \longmapsto u_r$. We can test for inherent instability (ill-conditioning) by calculating the absolute condition number (see sub-section 7.1.1),

$$\left| \frac{du_r}{dc} \right| = |a_r \cdots a_1|$$

Thus, a small error in the initial value c is magnified or diminished by a factor $|a_r \cdots a_1|$. If the magnitudes $|a_1|, |a_2|, |a_3|, \ldots$ are all greater than 1, this condition number may thus become very large as r increases, indicating that the problem is ill-conditioned.

Usually we are interested more in the relative than the absolute error, and the appropriate measure of ill-conditioning in this case is the relative condition number

$$\left| \frac{c}{u_r} \frac{du_r}{dc} \right| = \left| \frac{(a_r \cdots a_1)c}{(a_r \cdots a_1)c + \sum_{s=1}^{r} (a_r \cdots a_{s+1})b_s} \right|$$

$\left(\text{Notice that this is } \left| \frac{\text{element of kernel}}{\text{element of general solution}} \right| \cdot \right)$ Although this formula is more complicated than the last, we can still see the possibility of ill-conditioning: if $|a_r \cdots a_1|$ increases as r increases, while the numbers u_r we are trying to calculate increase less rapidly, or decrease, then the problem will be ill-conditioned for large r. In such a case, no matter how accurately we perform the subsequent computation, there will be a large

error in the computed result, both relatively and absolutely, and the problem is inherently unstable.

We can now explain the phenomena described in sub-section 7.1.5 relating to the computation of $I_r = e^{-1}\int_0^1 x^r e^x\,dx$ from a first-order recurrence relation which, to avoid ambiguities of notation, we write as

$$u_r = 1 - ru_{r-1}.$$

Here we have $a_r = -r$ and hence

$$\text{absolute condition number} = r!$$

$$\text{relative condition number} = \left|\frac{cr!}{u_r}\right|.$$

The relative condition number is greater than $r!$, since with $c = u_0$ taken as I_0, which is $1 - e^{-1}$, the sequence $u_0, u_1, u_2, \ldots$ can be shown to decrease. Thus for the values of r in the neighbourhood of 7, the relative condition number is somewhere around $7! = 5040$ and the problem is very ill-conditioned. This explains the enormous errors obtained when this method was used in the case study to calculate I_7 and similar integrals.

On the other hand, if we had been interested in the same recurrence relation but with $u_0 = 1$ (instead of $1 - e^{-1}$), the value of u_7 would have been -1754. In this case the relative condition number would have been $\frac{1 \times 5040}{1754}$, which is about 3, and so the problem of calculating u_7 by this method would have been reasonably well-conditioned. An error of $\pm 10^{-4}$ in the value of u_0 produces an absolute error of ± 0.5 in u_7 but a relative error of only about 1 part in 3000.

Turning now to the problem of induced instability, it is clear that the machine is likely to produce rounding errors in the successive computations of $u_1, u_2, \ldots$. For example, the exact formula for u_1 is

$$u_1 = a_1u_0 + b_1$$

but the floating-point machine computation will introduce an error e_1, so that the number actually stored by the machine is

$$\bar{u}_1 = a_1\bar{u}_0 + b_1 + e_1$$

where $\bar{u}_r$ is the stored value of u_r.

The next calculation, that of u_2, thus starts from the incorrect value $\bar{u}_1$ in place of u_1; moreover the floating-point arithmetic introduces a further error, say e_2, so that the number stored by the machine in place of u_2 is

$$\bar{u}_2 = a_2\bar{u}_1 + b_2 + e_2$$

Continuing in the same way we see that the numbers stored by the machine satisfy a recurrence relation

$$\bar{u}_r = a_r\bar{u}_{r-1} + b_r + e_r.$$

The general solution of this recurrence relation by the formula of Equation (2) of sub-section 7.2.2, is

$$\bar{u}_r = (a_r \cdots a_1)\bar{u}_0 + \sum_{s=1}^{r} (a_r \cdots a_{s+1})(b_r + e_r)$$

and thus it differs from the exact solution

$$u_r = (a_r \cdots a_1)u_0 + \sum_{s=1}^{r} (a_r \cdots a_{s+1})b_r$$

by an amount

$$(a_r \cdots a_1)e_0 + \sum_{s=1}^{r} (a_r \cdots a_{s+1})e_r,$$

where $e_0 = \bar{u}_0 - u_0$ is the error in the original data. Thus, if the numbers $|a_1|$, $|a_2|$, ... are greater than 1, and if r is large, then not only the original error e_0 but also the other errors e_1, e_2, etc., produced early on in the calculation will be magnified as the calculation proceeds. Thus in this problem the conditions which favour ill-conditioning (values of $|a_1|$, $|a_2|$, etc., exceeding 1) tend to favour induced instability as well. (In later units on numerical analysis we shall see examples where induced instability arises even in well-conditioned problems.) On the other hand if the numbers $|a_1|$, $|a_2|$, etc., are less than 1, then both the initial error in u_0 and the later ones e_1, e_2, ... will be diminished as the calculation proceeds to high values of r, and so the problem is likely to be both well-conditioned and free from induced instability.

The situation can be summarized in this way: if the solution we are looking for does not increase with r as fast as the solution of the homogeneous equation (the kernel), then for large r the solution we want will be swamped by an unwanted contribution from the kernel, stimulated either by errors in the data or by rounding errors from early stages of the calculation. On the other hand if the solution we want is increasing with r as fast as, or faster than, the unwanted contribution, then it can be calculated with small relative error, though perhaps large absolute error, by this method.

Exercise

Working to four significant figures, as in four-digit floating-point arithmetic, we have calculated $y_1, y_2, \ldots, y_6$ for the recurrence relation $y_{r+1} = 1 - (r + 6)y_r$, taking the initial value y_0 first as 0.15 and then as 0.16. The results are shown in the accompanying table. How much of the difference in the two values of y_6 is due to induced instability?

(*Hint* Can you calculate how much of it is *not* due to induced instability —that is, how much is due to inherent instability?)

	$y_0 = 0.15$		$y_0 = 0.16$	
r	y_r	$(r+6)y_r$	y_r	$(r+6)y_r$
0	0.1500	0.9000	0.16	0.96
1	0.1000	0.7000	0.04	0.28
2	0.3000	2.400	0.72	5.76
3	−1.4000	−12.60	−4.76	−42.84
4	13.60	136.0	43.84	438.4
5	−135.0	−1485	−437.4	−4811
6	1486		4812	

The difference of 0.01 in y_0 leads to a difference of 4812–1486 = 3326 in y_6.

Solution

To see how much of this difference arises from induced instability we compare it with the difference to be expected in an exact calculation. The exact solution of the recurrence relation is

$$y_r = (-1)^r(r + 5)(r + 4) \cdots (7)(6)c + \text{terms independent of } c$$

A change of 0.01 in c causes a change in the exactly computed value of y_6 amounting to

$$11 \times 10 \times 9 \times 8 \times 7 \times 6 \times 0.01 = 3326.4$$

This is the change in y_6 due to ill-conditioning alone. The rest of the change, which is due to induced instability, is $3326 - 3326.4 = -0.4$. In fact, because of the simple numbers used, the only rounding error in the whole calculation is in the last value of y_6 for $y_0 = 0.16$.

7.3.2 First-order Recurrence Relations—Backward Recurrence

In our case study of the evaluation of $I_r = e^{-1}\int_0^1 x^r e^x\,dx$ from the recurrence relation $I_r = 1 - rI_{r-1}$, we found that instead of starting from I_0 and working away from $r = 0$, it was much better to use the same recurrence relation but work towards $r = 0$. In other words we now regard the recurrence relation as giving I_{r-1} in terms of I_r, writing it in the form

$$I_{r-1} = \frac{1}{r}(1 - I_r).$$

Since the recurrence is now going the opposite way, it is reasonable to expect the errors to be diminished rather than magnified.

To start off the calculation with this form of the recurrence relation, we used the fact that I_r approaches zero for large r, and took as our starting point the approximation $I_n = 0$ for some sufficiently large value of n (about 14). Then we calculated successively I_{n-1}, I_{n-2}, and so on, and found that these numbers were surprizingly insensitive to the chosen values of n and I_n, indicating that this problem is very well-conditioned.

To remind you of the very good conditioning of this formulation of the problem of calculating I_r, we give some more numbers calculated from the backward recurrence relation, this time calculated working in floating-point arithmetic with *six* decimal digits.

r	u_r		
14		0	−1
13	0	0.071 428 6	0.142 857
12	0.076 923 1	0.071 428 5	0.065 934 1
11	0.076 923 1	0.077 380 9	0.077 838 8
10	0.083 916 1	0.083 874 5	0.083 832 8
9	0.091 608 4	0.091 612 6	0.091 616 7
8	0.100 932	0.100 932	0.100 931
7	0.112 384	0.112 384	0.112 384
6	0.126 802		
5	0.145 533		
4	0.170 893		
3	0.207 277		
2	0.264 241		
1	0.367 880		
0	0.632 120		

The computed u_0 agrees with the "required" value $1 - e^{-1}$ to all six figures.

The results confirm our expectations that the problem is well-conditioned, and that the errors decrease very rapidly as r decreases. By taking the three different starting values, and comparing the results, we see that the

three values of each of $u_8, u_7, \ldots, u_0$ are essentially equal, to six decimal figures, and though consistency is not always a good check, it would be very surprizing if it failed here! But in fact, as we shall see in a moment, we can perform a complete error analysis and specify in advance the value of n for which any particular u_r (with $r < n$), will have whatever precision we require.

To do the error analysis we shall consider the more general linear recurrence relation

$$u_r = a_r u_{r-1} + b_r$$

Provided all the a_r are non-zero, we can solve for u_{r-1} obtaining

$$u_{r-1} = \frac{u_r - b_r}{a_r}$$

To obtain the general solution starting with a particular value of u_n (with n large) we apply the recurrence relation with $r = n$, then $r = n - 1$, then $r = n - 2$, and so on. This gives

$$u_{n-1} = \frac{u_n - b_n}{a_n} = u_n/a_n - b_n/a_n$$

$$u_{n-2} = \frac{u_{n-1} - b_{n-1}}{a_{n-1}} = u_n/(a_n a_{n-1}) - b_n/(a_n a_{n-1}) - b_{n-1}/a_{n-1}$$

and, in general

$$\begin{aligned} u_r &= u_n/(a_n a_{n-1} \cdots a_{r+1}) - b_n/(a_n a_{n-1} \cdots a_{r+1}) \\ &\quad - b_{n-1}/(a_{n-1} \cdots a_{r+1}) - \cdots - b_{r+1}/a_{r+1} \\ &= u_n/(a_n \cdots a_{r+1}) - \sum_{s=r+1}^{n} b_s/(a_s \cdots a_{r+1}) \end{aligned}$$

This time we are using u_n to determine u_r, so that the absolute condition number is

$$\left| \frac{du_r}{du_n} \right| = \left| \frac{1}{a_n \cdots a_{r+1}} \right|$$

and the relative condition number is

$$\left| \frac{u_n}{a_n \cdots a_{r+1} u_r} \right|$$

If all the numbers $|a_1|, |a_2|, \ldots$ are greater than 1, then the absolute condition number will be small. Also, if the correct value of $|u_n|$ is less than $|u_r|$ for $r < n$, the relative condition number is even smaller. Thus, under these conditions, the problem is very well-conditioned.

These conclusions apply to the recurrence relation for the integral

$$I_r = e^{-1} \int_0^1 x^r e^x \, dx.$$

The recurrence relation $I_r = 1 - rI_{r-1}$ tells us that $a_r = -r$ and so the absolute condition number is

$$\left| \frac{1}{n(n-1) \cdots (r+1)} \right|$$

That is, any error in the initial value assumed for I_n is multiplied by this factor to give the numerical value of the error in the calculated value of I_r. Suppose for example, that we want I_{10} correct to 4 decimal places, that we start with $u_n = 0$ and (for the moment) that we make no rounding errors in the calculation. (It then follows, incidentally, that we also get I_9, I_8, I_7, etc. correct to 4 decimals.) By assuming $u_n = 0$ rather than

its correct value I_n, we make an error of magnitude I_n; and this leads to an error of magnitude

$$|e_{10}| = \left| \frac{I_n}{n(n-1)\cdots 12.11} \right|$$

in I_{10}. Now I_n can hardly exceed 0.5, since $I_0 = 1 - e^{-1} = 0.632\cdots$ and I_r decreases as r increases. The requirement that I_{10} be correct to 4 decimal places means that we want $|e_{10}| < \frac{1}{2} \times 10^{-4}$. Thus we want to choose n so that

$$\frac{0.5}{n \times (n-1)\cdots 12 \times 11} < \frac{1}{2} \times 10^{-4} = \frac{0.5}{10 \times 10 \times 10 \times 10}$$

By trying a few values for n if necessary, we can find that $n = 14$ (or any larger integer) satisfies this condition (for there are then 4 factors in the denominator on the left, all greater than 10, and 4 factors of exactly 10 on the right). So by using the approximation $I_{14} = 0$ we can guarantee (at least with exact arithmetic) that our values of I_{10}, I_9, I_8, etc., are correct to 4 decimal places.

Will the rounding errors of floating-point arithmetic produce any significant *induced* instability? As in the error analysis for forward recurrence, we can allow for these errors by replacing the recurrence relation by

$$u_{r-1} = \frac{u_r - b_r}{a_r} + e_r.$$

Solving this by the same method as before, gives

$$u_r = u_n/(a_n \cdots a_{r+1}) - \sum_{s=r+1}^{n} (b_s/(a_s \cdots a_{r+1}) + e_s/(a_{s-1} \cdots a_{r+1}))$$

$$= (\text{exact solution starting from } u_n) - \sum_{s=r+1}^{n} e_s/(a_{s-1} \cdots a_{r+1})$$

Thus the error in u_r due to induced instability is

$$\left| \frac{e_{r+1}}{1} + \frac{e_{r+2}}{a_{r+1}} + \cdots + \frac{e_n}{a_{n-1} \cdots a_{r+1}} \right|.$$

If all the $|a|$'s are greater than 1, the terms die away rapidly as we go along the series, and so most of the total effect of rounding error is little greater than the first term and this represents the rounding error introduced by the very last arithmetic operation in the calculation. Thus the conditions ($|a|$'s greater than 1) which make this problem so well-conditioned also make it free from induced instability. For example, in our case study, we are working to 4 significant figures with values of u_s that are less than 1; thus the errors $e_{r+1}, e_{r+2}, \ldots$ in the above formula do not exceed $\frac{1}{2} \times 10^{-4}$ and the entire formula (with $r = 10$) tells us that the error in u_{10} is at most

$$\frac{1}{2} \times 10^{-4}\left(1 + \frac{1}{11} + \frac{1}{12 \times 11} + \cdots\right) < \frac{1}{2} \times 10^{-4} \quad \left(1 + \frac{1}{10} + \frac{1}{100} + \cdots\right)$$

$$= \frac{1}{2} \times 10^{-4} \times \frac{1}{1 - \frac{1}{10}}$$

This formula tells us that the error in u_{10} due to the rounding is very little greater than the rounding error in the last step; i.e. in calculating

$$u_{10} = (1 - u_{11})/11$$

The main lesson to be learned from this error analysis is this: for a first-order recurrence relation, if the solution we are looking for grows less fast with increasing r than the solution of the homogeneous equation (the kernel), then it cannot be calculated by forward recurrence but it can be

calculated by backward recurrence; for as we *decrease* r, the unwanted contribution from the kernel *decreases* more rapidly than the solution we want.

Exercise

The integral $I_r = \int_0^{\pi/2} x^{2r} \cos x \, dx$ satisfies the recurrence relation

$$I_r = \left(\frac{\pi}{2}\right)^{2r} - 2r(2r-1)I_{r-1} \qquad (r = 1, 2, 3, \ldots)$$

with the initial condition $I_0 = 1$.

(i) Would you recommend forward or backward recurrence for calculating I_4 using 4-figure floating-point arithmetic? Justify your choice.
(ii) Show theoretically that the approximation $I_7 = 0$ is sufficiently accurate to give four-figure accuracy in the above calculation of I_4. (You may assume that

$$I_7 < \int_0^{\pi/2} x^{14} \, dx = \frac{1}{15}\left(\frac{\pi}{2}\right)^{15} < 60.)$$

Solution

(i) Backward recurrence is the recommended method. The solution of the homogeneous problem associated with the given recurrence relation

$$k_r = -2r(2r-1)k_{r-1},$$

is $k_r = (-1)^r(2r)!\,c$, which increases rapidly with r. To avoid unwanted contributions from elements of the kernel, we should recur in the direction in which this function decreases —i.e., backwards, in the direction of decreasing r.

(ii) Since the absolute condition number is

$$\left|\frac{du_r}{du_n}\right| = \left|\frac{1}{a_n \cdots a_{r+1}}\right|,$$

the absolute condition number in calculating I_4 form I_7 is

$$\left|\frac{1}{a_7 a_6 a_5}\right| = \frac{1}{(14 \times 13)(12 \times 11)(10 \times 9)} < \frac{1}{2} \times 10^{-6},$$

since $a_r = -2r(2r-1)$ in our recurrence relation. The error in replacing the true value of I_7 by 0 is at most $\frac{1}{15}\left(\frac{\pi}{2}\right)^{15} < 60$ and the consequent error in I_4 is at most

$$\tfrac{1}{2} \times 10^{-6} \times 60 = 30 \times 10^{-6} < \tfrac{1}{2} \times 10^{-4}$$

which is acceptable since only 4 places of decimals are wanted in I_4.

7.3.3 Second-order Recurrence Relations of Initial-value Type

We have seen that a second-order recurrence relation has a general solution which is the sum of a particular solution and the general solution of the associated homogeneous equation (the kernel). For example, we observed in part (iii) of the example of sub-section 7.2.3, that the equation

$$u_r = u_{r-1} + u_{r-2} + 1$$

has the general solution

$$u_r = -1 + c_1\left(\frac{1-\sqrt{5}}{2}\right)^r + c_2\left(\frac{1+\sqrt{5}}{2}\right)^r,$$

where $\left(\frac{1+\sqrt{5}}{2}\right)^r$ and $\left(\frac{1-\sqrt{5}}{2}\right)^r$ are independent solutions of the associated homogeneous equation $u_r = u_{r-1} + u_{r-2}$.

Such a solution has two arbitrary constants c_1 and c_2, which will be determined by the imposition of two more pieces of information about u_r. The most common are

(i) the specification of u_0 and u_1 (or of u_n and u_{n-1}) from which $u_2, u_3, \ldots$ (or $u_{n-2}, u_{n-3}, \ldots$) can be obtained uniquely by forward (backward) recurrence from the given recurrence relation; or

(ii) the specification of u_0 and u_n, from which we hope to determine the intermediate values $u_1, u_2, \ldots, u_{n-1}$.

Problem (i) is called an *initial-value problem*, and problem (ii) a *boundary-value problem*. Here we shall consider briefly some aspects of the initial-value problem, because they are quite similar to those we have discussed for the first-order recurrence relation problem.

From what we have learnt in the first-order case we can now easily see that the initial-value problem will be ill-conditioned whenever the required solution is less dominant than any of the elements of the kernel. Moreover in these circumstances induced instability will be manifest and will have the same nature as the inherent instability, just as in the first-order case.

Consider, for example, the *homogeneous* second-order relation

$$u_{r+1} - 10.1u_r + u_{r-1} = 0,$$

whose general solution is

$$u_r = c_1 10^r + c_2 10^{-r}$$

With the specification $u_0 = \pi$, $u_1 = \frac{\pi}{10}$, we find that $c_1 = 0$ and $c_2 = \pi$ so that we have theoretically suppressed the kernel* element 10^r, and should in theory obtain the solution $u_r = 10^{-r}\pi$ as a result of direct calculation. Some results obtained with four-figure floating-point arithmetic are

r	0	1	2	3	4	5
u_r	3.142	0.3142	0.0310	−0.001 100	−0.042 11	−0.4242

which not only "go wrong" very quickly but soon exhibit the tendency for successive values to exhibit growth by factors of 10, corresponding to the kernel element $c_1 10^r$.

Can we recover the situation by reformulation using backward recurrence, based on the knowledge that our required solution tends to zero as r gets larger? Since it is a second-order problem we need to specify both u_n and

* Note that $\{10^r, 10^{-r}\}$ forms a basis for the kernel.

u_{n-1}, and we take $u_n = 0$ which is a reasonable approximation for large n, and give u_{n-1} some non-zero value. The latter is probably not a good approximation, but if $\bar{u}_r$ is the computed solution of this problem then $k\bar{u}_r$ is also a solution (since the equation is homogeneous), and we can calculate a good value of k by making $k\bar{u}_0$ equal to the specified value of u_0. If k is small, then $k\bar{u}_{n-1}$ is also a reasonable approximation to the true value of u_{n-1}, and then the computed $k\bar{u}_r$ should be close to the required solution u_r for $r = 0, 1, 2, \ldots$ with small errors as r tends to n.

To introduce some rounding errors immediately we take $u_n = 0$, $u_{n-1} = e^{-1}$, and with $n = 5$ find results for the "trial" solution $\bar{u}_r$.

r	5	4	3	2	1	0
$\bar{u}_r$	0	0.3679	3.716	37.16	371.6	3716

Now multiplying this by $\text{fl}\left(\frac{3.142}{3716}\right) = 0.000\,845\,5$, in order to satisfy our given initial condition $u_0 = \pi$, we obtain the very good values

r	5	4	3	2	1	0
u_r	0	0.000 311 1	0.003 142	0.031 42	0.3142	3.142

There are only small absolute errors everywhere, even at $r = 5$, and only small relative errors almost everywhere, even as far away from the origin as $r = 4 = n - 1$ in this case.

We see that in practice the *homogeneous* second-order problem is very similar to the *non-homogeneous* first-order problem, and this is because in each case the solution falls into two distinct parts: in the first-order problem the parts are the element from the kernel and the particular solution, in the homogeneous second-order case the parts are the two independent solutions which form a basis for the kernel. We should perhaps remark that the analysis of a good starting point $r = n$ is here more involved, but the use of the "consistency" argument which we demonstrated on page 38 is in practice quite satisfactory.

In the homogeneous case our backward recurrence succeeded because the required solution, one of two possibilities, dominated in this direction. The non-homogeneous second-order problem, with three parts to its solution (the two independent solutions for the kernel plus the particular solution), produces a new situation in that the solution we want, one of *three* possibilities, may not dominate with recurrence in *either* direction.

This situation exists, for example, with the problem.

$$u_{r+1} - 10.1u_r + u_{r-1} = -2.7r,$$

with the initial conditions $u_0 = 0$, $u_1 = \frac{1}{3}$.

The general solution (see Exercise 3 of sub-section 7.2.3) is:

$$u_r = \tfrac{1}{3}r + c_1 10^r + c_2 10^{-r},$$

and the result of applying the initial conditions gives $c_1 = c_2 = 0$. But if we now begin at u_0 and make a direct numerical calculation, we know that using forward recurrence the term 10^r will take charge.

Forward recurrence

r	0	1	2	3	4	5	6
Calc. u_r	0	0.3333	0.6660	0.9940	1.274	1.076	−3.904
True u_r	0	0.3333	0.6667	1.0000	1.333	1.667	2.000

Further, using backward recurrence with the theoretically exact initial-value reformulation

$$u_{r+1} - 10.1u_r + u_{r-1} = -2.7r,$$

$$u_9 = 3, u_8 = \tfrac{8}{3},$$

we know that the term 10^{-r} will dominate. This is illustrated by the following catastrophic numerical results.

Backward recurrence

r	9	8	7	6	5	4	3
Calc. u_r	3	2.667	2.340	2.063	2.300	7.667	63.97
True u_r	3	2.667	2.333	2.000	1.667	1.333	1.000

Have we any other possible and this time successful reformulation? It turns out that whenever the initial-value problems in which either u_0 and u_1 or u_n and u_{n-1} are specified are both ill-conditioned, then the boundary-value problem with u_0 and u_n specified is likely to be well-conditioned.

It follows that if u_0 and u_1 are specified, then if we know some u_n from some independent consideration, the best formulation of the problem is to ignore u_1 and solve the boundary-value problem with u_0 and u_n given. In our case, for example, with $u_0 = 0$, $u_6 = 2$, we can write down the recurrence relations for $r = 1, 2, \ldots, 5$ as a set of linear simultaneous algebraic equations, in which the known u_0 and u_6 are transferred to the right-hand side. The equations are

$$\begin{aligned}
r = 1 \quad & -10.1u_1 + u_2 = -2.7 \\
r = 2 \quad & u_1 - 10.1u_2 + u_3 = -5.4 \\
r = 3 \quad & u_2 - 10.1u_3 + u_4 = -8.1 \\
r = 4 \quad & u_3 - 10.1u_4 + u_5 = -10.8 \\
r = 5 \quad & u_4 - 10.1u_5 = -15.5
\end{aligned}$$

In *Unit 8, Numerical Solution of Simultaneous Algebraic Equations*, we shall discuss methods for solving linear equations which are guaranteed to give good results whenever the problem is well-conditioned (that is, the methods have hardly any induced instability), and using such a method, with four-figure floating-point arithmetic, we actually produce values of u_r which are the correctly rounded versions of the exact solution.

You might ask what one can do if some u_n is not known, but that we do know that u_r tends to 0 as r increases. What we can do here is analogous to previous similar situations. We solve the relevant algebraic equations several times, taking $u_n = 0$ with different values of n, and assessing the

accuracy of our results by inspection. Taking, for example, $u_8 = 0$ and then $u_9 = 0$, we produce the results:

r	0	1	2	3	4	5
$(n = 8)u_r$	0	0.3334	0.6667	1.000	1.333	1.664
$(n = 9)u_r$	0	0.3334	0.6667	1.000	1.334	1.667

r	6	7	8	9
$(n = 8)u_r$	1.974	2.067	0	
$(n = 9)u_r$	1.997	2.304	2.367	0

They show how well-conditioned our boundary-value problem is, and also generate considerable confidence in the computed results at least up to $r = 5$.

We have now effectively finished our introduction to the treatment of second-order recurrence relations of initial-value. The only missing item is a full analysis, corresponding to that of sub-section 7.3.1 for the first-order case, for the maximum error in the computed u_r obtained with floating-point arithmetic for an initial-value problem in which the specified u_0 and u_1 might have small physical or mathematical uncertainties and in which rounding errors are induced in the subsequent arithmetic. This analysis is not part of this course, but you will find it on pages 40–42 of *Computing Methods for Scientists and Engineers* by Fox and Mayers (see Bibliography).

Exercises

For each of the following problems state:

(i) whether or not the problem is ill-conditioned;
(ii) how, in each ill-conditioned case, you might successfully reformulate it.

1. $2u_{r+1} - 5u_r + 2u_{r-1} = 0$; $u_0 = 0$, $u_1 = \pi$.
2. $2u_{r+1} - 5u_r + 2u_{r-1} = 0$; $u_0 = \pi$, $u_1 = \frac{1}{2}\pi$.
3. $2u_{r+1} - 5u_r + 2u_{r-1} = -\frac{1}{3}r$, $u_0 = 0$, $u_1 = 1$.
4. $2u_{r+1} - 5u_r + 2u_{r-1} = -\frac{1}{3}r$, $u_0 = 0$, $u_1 = \frac{1}{3}$.

(*Hint*: In 3 and 4 a particular solution is $u_r = \frac{1}{3}r$.)

Solutions

1. The general solution is $u_r = c_1 2^r + c_2(\frac{1}{2})^r$, so that the particular solution we want, satisfying the given initial conditions, is $u_r = \frac{2}{3}\pi(2^r - (\frac{1}{2})^r)$. This contains a large multiple of the dominant element of the kernel, so that the problem is relatively well-conditioned.
2. The required solution is $u_r = \pi(\frac{1}{2})^r$, but there is a more dominant kernel element 2^r. The problem is therefore ill-conditioned. Backward recurrence, with $u_n = 0$, $u_{n-1} = 1$, multiplied by a factor to give the correct $u_0 = \pi$, is a well-conditioned reformulation because the unwanted element is here decreasing and the required solution is increasing.

3. The general solution is $u_r = \frac{1}{3}r + c_1 2^r + c_2(\frac{1}{2})^r$, and with $u_0 = 0$, $u_1 = 1$ we find the solution

   $$u_r = \tfrac{1}{3}r + \tfrac{4}{9}(2^r - (\tfrac{1}{2})^r).$$

   This contains a significant contribution from the dominating element of the kernel, and the problem is relatively well-conditioned.
4. The required solution is $u_r = \frac{1}{3}r$. The problem is badly conditioned, and so is any initial-value reformulation (e.g. backward recurrence with $u_6 = 2$, $u_5 = \frac{5}{3}$). The reformulation as a boundary-value problem, as in the last part of sub-section 7.3.4 gives a well-conditioned problem, solvable without induced instability.

7.3.4 Summary of Section 7.3

In this section we defined the terms

forward recurrence	(page C 34)	* * *
backward recurrence	(page C 37)	* * *
initial-value problem	(page C 41)	* * *
boundary-value problem	(page C 41)	* * *

Techniques

1. For a given linear first-order or homogeneous second-order recurrence relation, analyse the inherent and induced instabilities. Where relevant avoid such instabilities by problem reformulation (e.g. recur in the reverse direction). * * *
2. For a given initial-value second-order linear recurrence relation, determine whether it is ill- or well-conditioned. In ill-conditioned cases, reformulate the problem (where possible) as a boundary-value problem. * * *

Notation

$\bar{u}_r$	(page C 35)

7.4 SUMMARY OF THE UNIT

In this unit we have discussed the following items.

1. The care needed in the formulation and solution of problems when numerical answers are required.
2. The difference between physical problems, in which some of the data are uncertain, and mathematical problems in which the data are exact but cannot be stored exactly in a computer.
3. The possibilities of *inherent* instability, so that few worthwhile figures can be quoted in the answer to a physical problem, and that if many figures are required in a mathematical problem (which is a meaningful request) much work might be involved.
4. The possibilities of *induced* instability caused by rapid accumulation of rounding errors in computer arithmetic.
5. The way the machine stores numbers and performs the four basic arithmetic operations in floating-point arithmetic.
6. The nature of linear recurrence relations, how they might appear in practice, and how to find some solutions for (i) general first-order equations and (ii) second-order equations with constant coefficients.
7. A full error analysis for the numerical solution of recurrence relations of first-order, revealing both inherent and induced instability. The possibility of avoiding both instabilities by problem reformulation; in particular, by recurring in the reverse direction. Similar treatment, without full error analysis, of homogeneous second-order recurrence relations with given initial conditions.
8. The impossibility of avoiding ill-conditioning for some solutions of certain non-homogeneous initial-value second-order equations and the importance of reformulation, where possible, as a boundary-value problem.

Definitions

The terms defined in this unit and page references to their definitions are given below.

ill-conditioned problem	(page C 7)	* * *
well-conditioned problem	(page C 7)	* * *
inherent instability	(page C 7)	* * *
magnification factor	(page C 7)	* *
absolute condition number	(page C 7)	* *
relative condition number	(page C 8)	* *
induced instability	(page C 10)	* * *
floating-point arithmetic	(page C 12)	* * *
relative error	(page C 12)	* * *
recurrence relation	(page C 22)	* * *
linear recurrence relation	(page C 22)	* * *
linear difference operator	(page C 29)	* *
order of a recurrence relation	(page C 30)	* * *
forward recurrence	(page C 34)	* * *
backward recurrence	(page C 37)	* * *
initial-value problem	(page C 41)	* * *
boundary-value problem	(page C 41)	* * *

Techniques

1. Use condition numbers to predict ill-conditioning. * *
2. Perform calculations in four-figure floating-point arithmetic and analyse the errors involved. * * *
3. Consider various methods for a given problem, choosing, if possible, the most stable and most economic one (as exemplified in the case study). * * *

4. Solve first-order linear recurrence relations and second-order linear recurrence relations with constant coefficients. * * *
5. For a given linear first-order or homogeneous second-order recurrence relation, analyse the inherent and induced instabilities. Where relevant avoid such instabilities by problem reformulation (e.g. recur in the reverse direction). * * *
6. For a given initial-value second-order linear recurrence relation, determine whether it is ill- or well-conditioned. In ill-conditioned cases, reformulate the problem (where possible) as a boundary-value problem. * * *

Notation

ε	(page C 6)
x	(page C 6)
X	(page C 6)
$\mathrm{fl}(x)$	(page C 12)
r_x	(page C 12)
r	(page C 14)
R^∞	(page C 28)
E	(page C 29)
$\bar{u}_r$	(page C 35)

7.5 SELF-ASSESSMENT

Self-assessment Test

This Self-assessment Test is designed to help you test quickly your understanding of the unit. It can also be used, together with the summary of the unit for revision. The answers to these questions will be found on the next non-facing page. We suggest you complete the whole test before looking at the answers.

1. For which values of x are the following problems ill-conditioned? (Use as your criterion of ill-conditioning a relative condition-number exceeding 1.)
   (i) Calculation of x^2.
   (ii) Calculation of $x^{1/2}$.
   (iii) Calculation of $\cos x$.

2. Explain in about 60 words the difference between induced and inherent instability.

3. (i) A number in the range [0, 100] is fed into a computer which uses four-digit floating-point arithmetic. What is the maximum absolute error in the stored value (the *inherent* "mathematical" error)?
   (ii) The stored values of two such numbers are $\bar{x}$ and $\bar{y}$. What are the additional (induced) maximum absolute errors in the computation of (a) $\bar{x} + \bar{y}$, (b) $\bar{x} - \bar{y}$?

4. The integrals $I_r = \int_0^1 x^{2r} e^{-x^2}\, dx$, for $r = 0, 1 \ldots$, satisfy the recurrence relation

$$I_r = -\frac{1}{2e} + \left(r - \frac{1}{2}\right) I_{r-1}.$$

   Describe, but do not carry out, a method of computing the value of I_0 to four figures which has reasonable accuracy using four-figure floating-point arithmetic.
   (*Hint* I_r decreases as r increases, and $I_0 \simeq 0.7$.)

5. Suppose that we have calculated I_0 (of Question 4) by some independent method, such as numerical integration using Simpson's rule, to an accuracy (largest possible error) of $\frac{1}{2} \times 10^{-4}$. If we now calculate I_6 by forward recurrence from the recurrence relation of Question 4, what is the resulting accuracy in the computed I_6, assuming that all the arithmetic in the computation from the recurrence relation is performed exactly (that is, there is only an inherent error in I_0 and no subsequent induced error)?

6. Find the general solutions of the recurrence relations
   (i) $u_{r+1} - 3u_r + 2u_{r-1} = 0$
   (ii) $u_r = u_{r-1} - 0.09u_{r-2} + 0.09$

7. We want to compute the solution $y = u_r$ of the recurrence relation

$$3u_{r+1} - 7u_r + 2u_{r-1} = 1 - 2r, \text{ for } r = 0, 1, \ldots, 20,$$

   and we have the following possible formulations of the problem.
   (a) Specification of u_0 and u_1.
   (b) Specification of u_{20} and u_{19}.
   (c) Specification of u_0 and u_{20}.
   Which formulation would you choose, and why?

Solutions to Self-assessment Test

1. For the calculation of $f(x)$, the relative condition number is defined as

$$k = |xf'(x)/f(x)|,$$

and if this exceeds 1 the problem is relatively ill-conditioned.

(i) $f(x) = x^2, f'(x) = 2x$, so that $k = 2$ and the problem is relatively ill-conditioned for any value of x.

(ii) $f(x) = x^{1/2}$, $f'(x) = \frac{1}{2}x^{-1/2}$, so that $k = \frac{1}{2}$ and the problem is relatively well-conditioned for any value of x.

(iii) $f(x) = \cos x, f'(x) = -\sin x$, so that $k = |x \tan x|$ and the problem is relatively ill-conditioned for those values of x for which $|x \tan x| > 1$.

2. *Induced instability* means that the *method of solution* is unsatisfactory, in that any rounding errors in the arithmetic accumulate to give a poor result even in a well-conditioned problem.

Inherent instability, or *ill-conditioning*, means that small changes in the data of the problem cause large changes in the solution. This is independent of the method of solution.

3. (i) The number is stored as $10^b \times a$, where a is a four-figure number $0.xxxx$ with first digit non-zero. So the maximum storage error is $10^b \times 0.5 \times 10^{-4}$. Since the number is smaller than 100, b is at most 2, so that the maximum absolute storage error is $0.5 \times 10^{-2} = 0.005$.

(ii) Since in floating-point arithmetic the sum or difference is formed to "double length" and then rounded to single length, the error is again just $10^b \times 0.5 \times 10^{-4}$, where the result of the operation is $10^b \times a$.

(a) In the addition the result may exceed 100, b might be 3, and the maximum error is then $0.5 \times 10^{-1} = 0.05$.

(b) In subtraction, both numbers being positive, the result is smaller than 100, so that $b \leqslant 2$ and the maximum rounding error is 0.005.

4. Since I_r decreases as r increases, and since the kernel element, $\frac{1}{2} \times \frac{3}{2} \times \cdots \times (r - \frac{1}{2})$, increases as r increases, we expect to be able to compute I_0, starting with any I_n for large n, and recurring backwards with

$$I_{r-1} = \frac{I_r + \frac{1}{2}e^{-1}}{r - \frac{1}{2}}.$$

Rounding errors do not accumulate, and no special arithmetic precautions are necessary, so that a four-figure approximation to $\frac{1}{2}e^{-1}$ should suffice for four-figure accuracy everywhere.

Apart from rounding errors, the error in the computed I_0 is the multiple

$$\frac{1}{(n - \frac{1}{2})(n - \frac{3}{2}) \cdots (\frac{1}{2})}$$

times the error in I_n. Since $I_0 \simeq 0.7$ and I_r decreases with r, the error in I_n can hardly exceed 0.7. For the error from this source to be less than 5×10^{-5} (for four-figure accuracy) we therefore choose n so that

$$\frac{0.7}{(n - \frac{1}{2})(n - \frac{3}{2}) \cdots (\frac{1}{2})} < 5 \times 10^{-5}.$$

The major rounding error is in the last step, with $r = 1$, and is therefore

twice the error in I_1 + the error in the stored value of e^{-1},

contributing at most $1\frac{1}{2}$ units in the fourth figure of I_0.

5. If we know I_0 with a possible error of ε, then the possible error in I_6, from this source alone (that is with no further mistakes in the arithmetic) is

$$\frac{1}{2} \times \frac{3}{2} \times \cdots \times \frac{11}{2} \varepsilon \leqslant \frac{10\,395}{128} \times 10^{-4} \leqslant 0.01.$$

So there might be an error in the second decimal place of I_6.

6. (i) The general solution of

$$u_{r+1} - 3u_r + 2u_{r-1} = 0$$

is $c_1 p_1^r + c_2 p_2^r$, where p_1 and p_2 are the roots of the quadratic equation $p^2 - 3p + 2 = 0$. That is, $p_1 = 2, p_2 = 1$, and the general solution is

$$u_r = c_1 2^r + c_2,$$

where c_1 and c_2 are constants.

(ii) The general solution of

$$u_r = u_{r-1} - 0.09u_{r-2} + 0.09$$

consists of a particular solution plus the solutions in the kernel. The latter are $c_1 p_1^r + c_2 p_2^r$, where p_1 and p_2 are the roots of the quadratic equation

$p^2 - p + 0.09 = 0$, that is, $p_1 = 0.9, p_2 = 0.1$.

For a particular solution, examine whether $u_r = c_3$, a constant, is a solution. Substitution gives

$c_3 - c_3 + 0.09c_3 = 0.09$,

so that $c_3 = 1$ is a solution, and the general solution is therefore

$u_r = c_1(0.9)^r + c_2(0.1)^r + 1$.

7. The solutions in the kernel are $c_1 p_1^r + c_2 p_2^r$, where p_1 and p_2 are the roots of the quadratic equation $3p^2 - 7p + 2 = 0$, that is $p_1 = 2$, $p_2 = \frac{1}{3}$. If a particular solution is r, the general solution is

$$u_r = c_1 2^r + c_2(\tfrac{1}{3})^r + r.$$

(a) If u_0 and u_1 are specified, we must use forward recurrence, and the unavoidable introduction of the term $c_1 2^r$ (through rounding errors) will swamp the less rapidly increasing required solution.

(b) If u_{20} and u_{19} are specified, we must use backward recurrence, and the other solution $c_2(\frac{1}{3})^r$ will now dominate.

(c) If u_0 and u_{20} are specified, we can solve the algebraic equation obtained from the recurrence relations with $r = 1, 2, \ldots, 19$, and we have a known stable method for their solution.

Formulation (c) is therefore preferred.

LINEAR MATHEMATICS

1 Vector Spaces
2 Linear Transformations
3 Hermite Normal Form
4 Differential Equations I
5 Determinants and Eigenvalues
6 NO TEXT
7 Introduction to Numerical Mathematics: Recurrence Relations
8 Numerical Solution of Simultaneous Algebraic Equations
9 Differential Equations II: Homogeneous Equations
10 Jordan Normal Form
11 Differential Equations III; Nonhomogeneous Equations
12 Linear Functionals and Duality
13 Systems of Differential Equations
14 Bilinear and Quadratic Forms
15 Affine Geometry and Convex Cones
16 Euclidean Spaces I: Inner Products
17 NO TEXT
18 Linear Programming
19 Least-squares Approximation
20 Euclidean Spaces II: Convergence and Bases
21 Numerical Solution of Differential Equations
22 Fourier Series
23 The Wave Equation
24 Orthogonal and Symmetric Transformations
25 Boundary-value Problems
26 NO TEXT
27 Chebyshev Approximation
28 Theory of Games
29 Laplace Transforms
30 Numerical Solution of Eigenvalue Problems
31 Fourier Transforms
32 The Heat Conduction Equation
33 Existence and Uniqueness Theorem for Differential Equations
34 NO TEXT